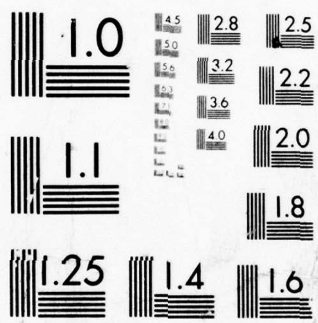
AD-A052 561   CARNEGIE INST OF TECH PITTSBURGH PA DEPT OF CIVIL EN--ETC F/G 12/1
SYMBOLIC GENERATION OF FINITE ELEMENT STIFFNESS MATRICES.(U)
JAN 78 A R KORNOFF, S J FENVES   N00014-76-C-0345

UNCLASSIFIED   R-78-103   NL

1 OF 2
AD
A052561



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SYMBOLIC GENERATION OF FINITE ELEMENT STIFFNESS MATRICES

Alan R. Korncoff and Steven J. Fenves

A Technical Report of a Research Program
Sponsored by
The Office of Naval Research
Department of the Navy
Contract N00014-76-C-0345

January 1978

DEPARTMENT OF CIVIL ENGINEERING
CARNEGIE INSTITUTE OF TECHNOLOGY
Carnegie-Mellon University

103
R-78-[redacted]

6

SYMBOLIC GENERATION OF FINITE ELEMENT STIFFNESS MATRICES.

9 Technical rept.

10 Alan R. Korncoff

and

Steven J. Fenves

11 Jan 78

12 118 p.

14 R-78-1Ø3

A Technical Report of a Research Program

Sponsored by

The Office of Naval Research

Department of the Navy

Contract NØØ014-76-C-Ø345

15

ABSTRACT

A symbolic processor, ***** (pronounced "five star"), to assist in the generation of stiffness matrices for finite elements, based on a recently developed symbolic processor, is presented. Operations are performed upon element characteristics and material properties in symbolic form to produce a "matrix template," consisting of the algebraic expressions generated for the stiffness coefficients as functions of the problem parameters in literal form. The template may be evaluated for a given element by binding these symbolic forms to the numerical values associated with a specific element. The evaluation process is further facilitated by permitting specification of a variety of output formats for the resulting matrix template. Required input is minimized by automatically synthesizing the constituent matrices of the formulation from user-supplied specifications of shape functions, material properties and stress-strain relationships, all in symbolic notation.

The processor, written in MACSYMA, is highly interactive providing prompts for user input, enumeration of available program options, and extensive on-line assistance. The user may input a "?" in place of a prompted input to request instructional text. The file handling capabilities of MACSYMA are utilized to retain a complete record of each program run. These records facilitate the handling of diagnostics, assist in further processing and permit the generation of statistics valuable for system development. Error checking is accomplished through semantic checks built into the program functions and syntactic checks performed within the MACSYMA operating environment.

A partial list of user input includes:

1) Method Selection - Isoparametric or generalized coordinate formulations. 2) Element Parameters - Number of nodes, number of degrees of freedom per node and related terms. 3) Material Properties - This matrix may be selected from a library of standard forms (e.g. plane stress, plane strain) or supplied by the user. 4) Strain Specification - Components are entered in a user-oriented calculus notation (e.g. $\partial u/\partial x$ is input as D(u,x)). 5) Shape Functions - Shape functions may contain trigonometric functions and a large class of intrinsic functions as well as polynomial terms. 6) Output Control Specification - A description of the output format of the generated matrix template.

Possible output forms include a tabular display of the matrix coefficients in symbolic form and the coefficents in FORTRAN card image format.

Background material includes: The objective of this study; The derivation of the stiffness matrices; A summary of previous research; A brief description of MACSYMA.

Details of the implementation of ***** cover: The design objectives; Details of the algorithms used and how they were implemented; A description of ***** and its limitations.

Sample runs include the formulation, using both the isoparametric and generalized coordinate methods, of the stiffness matrices for a: Bar element with constant cross-sectional area; Bar element with linearly varying cross-sectional area; Constant Strain Triangle with uniform thickness; Four Node Quadrilateral.

Conclusions are drawn and recommendations for future work are made. Appendix I contains notes on operating and accessing the processor.

ACKNOWLEDGEMENTS

This report was prepared as a doctoral dissertation by Dr. Alan R. Korncoff and was submitted to the Department of Civil Engineering of Carnegie - Mellon University in partial fulfillment of the requirement for the degree of Doctor of Philosophy in Civil Engineering. The work was conducted under the supervision of Dr. Steven J. Fenves, University Professor of Civil Engineering.

Gratitude is also extended to the other members of the author's doctoral committee: Prof. Tung Au, Prof. Charles Eastman, Prof. Charles H. Goodspeed and Prof. J. L. Swedlow for their assistance and encouragement.

Thanks are also due Mr. Jeffrey P. Golden, of the Laboratory for Computer Science, Massachusetts Institute of Technology, for his patience and assistance in resolving many of the author's questions regarding MACSYMA.

TABLE OF CONTENTS

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Objective

The finite element method involves two processes: the generation of elements, where the computation effort is linear in the number of elements, and the solution of the discrete system equations, where the effort increases as some power of the number of degrees of freedom, and thus of the number of elements. Improvements over the past decade in decomposition and solution techniques have reached the point where in many problems the element generation effort exceeds that for system solution. Thus, from a practical standpoint there is a great incentive to attempt to drastically reduce the computational effort in element generation, the bulk of which is taken up in the numerical quadrature involved. With the large variety of new elements being developed or investigated, there is a similar incentive to reduce the amount of manual algebraic manipulations required before the stiffness matrices for a new element can be cast in a form suitable for processing.

This research project was undertaken to develop a mechanism that would facilitate the generation of element stiffness matrices by eliminating the numerical quadrature process and by minimizing the amount of manual algebraic manipulation required.

A symbolic processor, [illegible], to assist in the generation of stiffness matrices for finite elements, based on a recently developed symbolic processor, is presented. Operations are performed upon element characteristics and material properties in symbolic form to produce a "matrix template," consisting of the algebraic expressions for the stiffness coefficients as functions of the problem parameters in literal form. The template may be evaluated for a given element by binding these symbolic forms to the numerical values associated with a specific element. The evaluation process is further facilitated by permitting specification of a variety of output formats for the resulting matrix template. Required input is minimized by automatically synthesizing the constituent matrices of the formulation from user-supplied specifications of shape functions, material properties and stress-strain relationships, all in symbolic notation.

1.2 Derivation of the Stiffness Matrix

This section introduces the nomenclature, details the required operations and contains the derivation of stiffness matrices for both the isoparametric and generalized coordinate formulations.

The stiffness matrix for any finite element is given, in general, by (see Desai and Abel[9], Zienkiewicz[10]):

$$[K] = \int [B]^t [C] [B] dV \quad (1)$$

The terms in equation (1) are discussed below.

1.2.1 Isoparametric Formulation

For isoparametric elements equation (1) is rewritten in terms of natural coordinates. In three dimensions this produces

$$[K] = \iiint [B]^t [C] [B] \det([J]) \, dr \, ds \, dt \quad (2)$$

in which [K] is the stiffness matrix and r, s and t are natural coordinates (Desai and Abel[9]). The derivation of the constituent matrices of equation (2) is given below.

In the isoparametric formulation, the functional relationship describing the element geometry and the element displacement are the same:

$$\{x\} = \{ N(r,s,t) \} \{Xn\} \quad (3)$$

$$\{u\} = \{ N(r,s,t) \} \{q\} \quad (4)$$

Here {x} represents the cartesian coordinates of the element; {Xn} is the vector of nodal coordinates in the global coordinate system; {u} stands for the values of the displacements interior to the element; {q} represents the nodal displacements and {N(r,s,t)} are the interpolation functions in terms of the non-dimensional natural coordinates. (Although the present development assumes a global cartesian system, a similar derivation can be written for other coordinate systems). The dimensionality of the above relationship is clarified in the following expansion of equation (4). Letting m = the number of nodes,

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_{3\times1} = \begin{bmatrix} \{N(r,s,t)\}_{1\times m} & 0 & 0 \\ 0 & \{N(r,s,t)\}_{1\times m} & 0 \\ 0 & 0 & \{N(r,s,t)\}_{1\times m} \end{bmatrix} \begin{Bmatrix} \{q_u\}_{m\times1} \\ \{q_v\}_{m\times1} \\ \{q_w\}_{m\times1} \end{Bmatrix} \quad (5)$$

The next step is to determine the strains which are derivatives of the displacements:

$$\{e\} = [B] \{q\} \quad (6)$$

in which {e} represents the strain components defined in the global coordinate system; and [B] represents the derivatives of the interpolation functions in equation (4) with respect to the global coordinates.

Since these interpolation functions (and hence, the displacements) are functions of the natural coordinates, differentiation must be performed by the chain rule. Employing the derivative of equation (3) with respect to each natural coordinate in the expression for the chain rule produces:

$$\begin{Bmatrix} \left\{\frac{\partial N}{\partial r}\right\} \\ \left\{\frac{\partial N}{\partial s}\right\} \\ \left\{\frac{\partial N}{\partial t}\right\} \end{Bmatrix}_{3\times m} = [J]_{3\times 3} \begin{Bmatrix} \left\{\frac{\partial N}{\partial x}\right\} \\ \left\{\frac{\partial N}{\partial y}\right\} \\ \left\{\frac{\partial N}{\partial z}\right\} \end{Bmatrix}_{3\times m} \qquad (7)$$

in which the Jacobian, [J] , is defined by

$$[J] = \begin{Bmatrix} \left\{\frac{\partial N}{\partial r}\right\} \\ \left\{\frac{\partial N}{\partial s}\right\} \\ \left\{\frac{\partial N}{\partial t}\right\} \end{Bmatrix}_{3\times m} \left\{ \{Xn\} \quad \{Yn\} \quad \{Zn\} \right\}_{m\times 3} \qquad (8)$$

As the quantity on the right side of equation (7) is required in the formulation of {B} , the Jacobian must be inverted. Premultiplying both sides of equation (7) by the inverse of the Jacobian produces an expression for the derivatives of the shape functions with respect to the global coordinates. These derivatives are assembled into the {B} matrix and ordered according to the specifications given by the strain component vector, {e}.

The [C] matrix contains the stress-strain relationships:

$$\{s\} = [C]\ \{e\} \qquad (9)$$

Finally, when the volume integral is converted from global to natural coordinates, the differential volume becomes

$$dx\ dy\ dz = \det([J])\ dr\ ds\ dt \qquad (10)$$

in which [J] is the Jacobian as defined in equation (8).

1.2.2 Generalized Coordinate Formulation

The generalized coordinate formulation begins with a relation expressing {u}, the displacements internal to the element as a function of a set of yet to be determined generalized coordinates represented by the column vector, {a}:

$$\{u\} = \{S(x)\}\ \{a\} \tag{11}$$

The shape functions, {S}, are polynomials in the global coordinates (denoted by x) and are chosen to conform to convergence requirements. In general, the order of the polynomials is such that the number of generalized coordinates is equal to the total number of degrees of freedom of the element. Utilizing this principle, the displacements interior to the element may be expressed in terms of the nodal displacements as follows. Substituting the nodal coordinates into the shape functions produces the displacement transformation matrix, [A], which relates the nodal displacements, {q}, to the generalized coordinates:

$$\{q\} = [A]\ \{a\} \tag{12}$$

Solving for {a}

$$\{a\} = [A]^{-1}\ \{q\} \tag{13}$$

and substituting into equation (11) produces the desired relationship between nodal and element displacements

$$\{u\} = \{S(x)\}\ [A]^{-1}\ \{q\} \tag{14}$$

This form is fundamentally the same as equation (4) for the isoparametric case.

For a specified set of strain components, the {B} matrix for the generalized coordinate approach, {Ba}, may be determined by appropriate differentiation of equation (11).

$$\{e\} = \{Ba\}\ \{a\} \tag{15}$$

Since the strains are derivatives of the displacements with respect to the global coordinates and the shape functions are polynomials in these coordinates, no coordinate system transformation, as in equation (7) for the isoparametric case, is required.

The material properties matrix, [C], is the same as in the isoparametric method.

Substituting the constituent matrices into equation (1) and integrating with respect to the differential volume, dV, in terms of the global coordinates, produces a stiffness matrix, [Ka], associated with the generalized coordinates.

$$[Ka] = \int \{Ba\}^t \, [C] \, \{Ba\} \, dV \qquad (16)$$

Equation (13) gives the transformation necessary to express the stiffness matrix with respect to the nodal displacement quantities:.

$$[K] = [A]^{-1\,t} \, [Ka] \, [A]^{-1} \qquad (17)$$

Equations (2) and (17) are the expressions for the stiffness matrix employed in the computations perfomed by the processor.

1.2.3 Formulation of Element Loads, Mass and Stresses

In addition to the stiffness matrix, several other element matrices are relevant to finite element analysis. These matrices include the element mass matrix, the stress - displacement relationships and the load vectors representing (a) body forces, (b) surface tractions and (c) initial strains.

Using the nomenclature introduced in section 1.2.1 and letting {L} represent the shape functions for either the isoparametric or generalized coordinate methods, the formulation of the above matrices follows:

The element body force vector, {QB} is given by

$$\{QB\} = \int \{L\}^t \, \{fb\} \, dV \qquad (18)$$

in which {fb} represents the specified body forces. If {L} contains the generalized coordinate shape functions, {QB} must be pre-multiplied by the inverse - tranpose of the [A] matrix of equation (12).

The element surface traction vector, {QT}, is given by

$$\{QT\} = \oint \{L\}^t \, \{ft\} \, dV \qquad (19)$$

in which {ft} represents the specified surface tractions and integration is performed over the surface of the element. The transformation involving the [A] matrix would be applied as in the formulation of {QB}.

The element initial force vector, {QI}, is given by

$$\{QI\} = \int \{B\}^t [C] \{ei\} \, dV \tag{20}$$

in which {B} stands for the {B} matrix of equation (6) or the {Ba} matrix of equation (15) and [C] is defined in equation (9). The vector of initial strains, due for example to thermal effects or misfit, is represented by {ei}. The transformation involving the [A] matrix is applied as in the formulation of {QB} and {QT}.

The element mass matrix, [M], is defined by

$$[M] = \int \{L\}^t [m] \{L\} \, dV \tag{21}$$

in which [m] is the specified mass density per unit volume tensor. If {L} represents the the generalized coordinate shape functions [M] must be pre- and post-multiplied by the inverse - transpose of [A] and the inverse of [A] repspectively.

The solution of the finite element problem produces the vector of nodal displacments, {q}. The element stresses, {s} may be calculated from

$$\{s\} = [C] \{B\} \{q\} \tag{22}$$

Issues in the implementation of the above quantities are discussed in section 2.9.

1.3 Motivation for Symbolic Processing

Numerical integration techniques require the evaluation of the integrand of equation (1) at specified points, producing a square matrix of numerical values of order equal to the number of degrees of freedom. As constituent terms in this equation are functions of the nodal coordinates (or are geometric values which must be evaluated at these points) the evaluation must be performed on each element individually. If a large number of elements is to be employed in the finite element model, a significant computational overhead may be incurred.

Similarly, numeric evaluation of the inverse and determinant of the Jacobian (equation (7)) and the inverse of the displacement transformation matrix (equation (13)) also requires nodal coordinates to be bound to numeric values and thus must be performed upon elements individually.

Using symbolic manipulation techniques, nodal coordinates may be retained and operated upon in literal form throughout the computation. The coordinates are not bound to numerical values and thus may represent any set of actual (numerical) coordinates. The stiffness matrix produced is expressed as a template in terms of these unbound values. It may be readily evaluated for a given set of actual nodal coordinates during execution. The computations detailed in the previous section need only be performed once to generate this general template.

1.4 Summary of Previous Work

Several researchers have proposed or investigated the use of symbolic computing languages for generating finite element stiffness matrices directly in literal form, to be subsequently evaluated for the specific numerical values of a given element.

Luft{1} proposes individual special purpose routines with limitations as to problem type, element shape, and displacement function specification. Two example programs are provided. The discussion for the first, a processor for rectangular elements using generalized coordinates, indicates that exact integration is being performed because the element boundaries correspond to constant values of X and Y. A program employing an isoparametric formulation comprises the second example. Since it utilizes Gauss Quadrature and the numerical evaluation of the Jacobian, it must be executed for each element individually. Contributions detailed include: (a) the introduction of 'intrinsic matrices' (the integral of the product of a form of the interpolation functions with its transpose) for minimizing and organizing intermediate calculations and; (b) the detailed specification of a polynomial manipulator for performing the requisite operations.

The processor descibed in Gunderson{2} employs a modified generalized coordinate approach and requires the user to specify the displacement transformation matrix (i.e. the inverse of the matrix relating the generalized coordinates to the nodal displacements). The program makes use of a sophisticated scheme for data organization by representing the polynomials as multi-dimensional integer arrays and by defining the matrix operations accordingly.

Taig{3} has presented programs for evaluating the stiffness coefficients for quadrilateral plate elements with in-plane forces for the cases of rectangular and trapezoidal panels.

Anderson and Noor{4} and Anderson and Bowen{5} demonstrate the use of MACSYMA's symbolic integration facilities in the development of shallow-shell finite elements. In both papers the required integrals are divided into classes using group theoretic techniques. In reference {5}, the symbolic expressions are incorporated into a FORTRAN computer program. A study is made of computation time and memory requirements.

Wong{6} suggests the use of symbolic computation in the formulation of finite element stiffness matrices but admits that CONFORM, the system that he co-authored, is insufficiently flexible for the required handling of data structures.

INTRODUCTION

The survey in Jensen[13] of symbolic computing languages and their applications in mechanics suggests that only MACSYMA possess sufficient flexibility and power to carry out the objectives.

In summary the work to date has been directed towards the development of either: (a) programs that take advantage of properties associated with a very specific element type ([3], [4], [5]); or (b) processors which synthesize matrices for any element but cannot automate the entire process ([1], [2]).

1.5 Description of MACSYMA

This section contains a brief description of the capabilities of MACSYMA, the base language of *****, and will serve as background material for subsequent dicussion of the implementation of the system.

MACSYMA (pronounced "maxima"), Project MAC's SYmbolic Manipulation system, is a large computer program written in LISP devoted to the manipulation of algebraic expressions. MACSYMA runs under the ITS timesharing system (originally developed at the M.I.T. Artificial Intelligence Laboratory), on the Mathlab PDP-10 computer at M.I.T. With a syntax resembling ALGOL 60, MACSYMA has capabilities for manipulating algebraic expressions involving constants, variables and functions [1].

Provisions and attributes of MACSYMA of interest to the computer language designer include:

(a) The base language is a specially designed, enriched version of LISP called MACLISP;

(b) The system contains a large body of intrinsic functions;

(c) ALGOL-like control structure, compound statements and block structure permit the incorporation of the intrinsics into user-defined functions. As in LISP, all functions, user-defined or intrinsic, return a value;

(d) A set of commands may be pre-stored on a disk file and executed by means of the 'BATCH' command;

(e) An editor, modeled after TECO, can be invoked to edit input or to correct syntax errors;

(f) Data types include atomic variables, lists, arrays, matrices, and strings. Numeric constants may be integers, rational number, floating point numbers or "bigfloats" (floating point values of essentially arbitrary precision);

(g) Debugging aids and trace functions are provided;

(h) The user may declare and manipulate properties of atoms;

(i) Pattern matching facilities exist. These include type testing and general pattern matching functions which permit the user to test expressions for combinations of syntactic and semantic patterns and to automatically have variables set to parts of the expressions which fit the patterns [6];

(j) It is possible to store expressions, values and functions on disk files.

Mathematical and computational functions include:

(a) Evaluation and simplification;

(b) Differentiation and integration;

(c) Part selction and substitution;

(d) Solving for roots of an equation;

(e) Matrix functions including transpostion, multiplication, inversion and evaluation of determinants.

(f) Manipulation of rational expressions (expressions which are the quotient of two polynomials);

(g) Taylor series and power series;

(h) Graphing;

(i) Poisson series;

(j) Tensor manipulation;

(k) Laplace transforms;

(l) Finding the limit of an expression as a constituent variable approaches a value from a given direction.

Command lines to MACSYMA are strings of characters representing mathematical expressions involving equations, arrays, functions, and programs. Extra spaces, tabs, and all carriage returns are ignored (except when these occur in quoted strings) [2].

Command lines are terminated by ";" or "$" (dollar sign). A ";" causes the command line to be evaluated and the result dispayed . The terminator "$" causes the command line to be evaluated but the result is not displayed [3]. As ***** supervises the display of output, both terminators have the same effect.

The command (input) lines are indexed by labels of the form "(Ci)" where i is incremented with each new command typed by the user. Similarly, the results of computations are also indexed by a label of the form "(Di)"; thus, usually the ith input-output pair will be(Ci)-(Di) [5]. Intermediate results (if any) are tagged with line labels of the form "(Ei)". Line labels may also be used to reference associated expressions.

Bogen [7] contains a description of the major features and functions of MACSYMA. Mathlab [8] provides details on logging into the system and contains a script.

An important characteristic is that MACSYMA, in its present, experimental version, provides a significantly limited amount of user storage space for symbolic expressions.

1.6 Organization of This Report

The remainder of this report is organized as follows:

Chapter 2 describes the system implementation, including the Design Consideration and Constraints; Details of the implementation of the computation of the stiffness matrix; and constituent matrices; Specification of algorithms; Logic Hierarchy and Data Flow; Unique Input and Output Facilities; Error Recovery; and System Limitations.

Chapter 3 contains a number of illustrative examples of runs made on the system and the verification of results for several test cases.

Chapter 4 consists of a summary and conclusions giving a final assessment of the work performed and suggestions for future research.

Appendix I is a brief user's manual to MACSYMA and *****.

Chapter 2

SYSTEM IMPLEMENTATION

2.1 Implementation Considerations

The considerations in the design of ***** were imposed by the operating environment of MACSYMA and a set of initial design decisions. The major design decisions were as follows:

(a) The system had to be easily usable by engineers possessing a minimum knowledge of its operation.

(b) Ease of maintenance and modification were be emphasized. The incorporation of improved algorithms and additional methods of analysis would thus be facilitated.

(c) The modularization of the system was intended to resemble the steps in the problem formulation.

(d) Consistent with MACSYMA's interactive mode of operation, the system would operate as an interactive process. Techniques which provide selective ouput were necessary to override MACSYMA's procedure of echoing commands and output.

(e) Since computation time increases with the number and complexity of the expressions, in general, there is no "compute versus store" tradeoff in MACSYMA. Processes may also become storage bound with respect to invoked program units, both user-defined and intrinsic. Whereas user program modules can be deleted in order to free storage, MACSYMA program segments, once loaded, remain for the duration of the job. A constraint was also placed upon computation time because the system is to be accessed interactively.

(f) All input had to be format-free.

(g) Semantic and syntactic checks on input had to performed. Integration of MACSYMA's syxtax checker and editor was deemed appropriate.

(h) It was decided that a copy should be retained of each session during which the system is used. Such records would aid in diagnostics and development.

(i) As the system is to run interactively, a help facility had to be provided to supplement error messages generated by semantic checks as well as to assist in entering input.

As an alternative to the general method of computation, which produces the complete matrix template by performing all integrations and matrix operations symbolically, a hybrid symbolic-numeric scheme was also considered. In this approach, symbolic operations are used only to produce the template for the triple product integrand of equation (1) (section 1.2), in terms of unbound values of the coordinates at the numeric quadrature points. At execution time of the analysis program invoking it, the template is numerically evaluated at each quadrature point to produce the integral. The hybrid technique would use the same computation sequence as either the isoparametric or generalized coordinate formulations, up to the symbolic integration processing.

2.2 Computation Implementation

Algorithms employed to perform the computations described in section 1.2 are presented.

2.2.1 Reformulation of the Stiffness Matrix

To minimize computational effort, the formulation of the stiffness matrices given in equation (2) for the isoparametric method and equation (17) for the generalized coordinate approach may be reorganized as follows.

For the isoparametric formulation, expressing the inverse of the Jacobian, [J], of equation (8) as the adjoint divided by the determinant, the latter may be factored from {B} and the transpose of {B}. Combining these terms with the determinant from equation 10 and using {BJ} to denote the reduced {B} matrix, yields the following expression for the stiffness matrix

$$[K] = \int \frac{1}{\det([J])} \{BJ\}^t \, [C] \, \{BJ\} \, dr\, ds\, dt \qquad (23)$$

This formulation produces a greatly simplified {B} matrix and reduces the number of required divisions in the finished matrix template.

A similar procedure is employed to factor the [A] matrix in the generalized coordinate method. Letting [AD] represent the determinant of the submatrices comprising the [A] matrix and [AJ] stand for the form of the [A] matrix produced by replacing these submatrices with their adjoint, equation (17) may be rewritten as

$$[K] = \frac{1}{\det([AD])^{**}2} \, [AJ]^t \, [Ka] \, [AJ] \qquad (24)$$

Once again the computations are greatly simplified.

2.2.2 Reduction of Dependent Natural Coordinates

The number of natural coordinates specified by the user must be equal to or, at most, one greater than the number of global coordinates specified. In the latter case, the natural coordinates are not independent.

To facilitate computation the last coordinate is treated as the dependent one and is replaced by an expression equal to unity minus the sum of the remaining coordinates.

2.2.3 Generation of the Jacobian

The Jacobian, as defined in equation (8), is formed using a refined version of the following algorithm.

Let

NDOF = the number of degrees of freedom per node,
NN = the number of nodes,
G[i] = the name of the ith global coordinate,
L[i] = the name of the ith natural coordinate,
S = the vector of shape functions,
DIFF, GC and JVAL are local values.

Then

```
1  for i=1 to NDOF do
2      DIFF=(the derivative of S with respect to L[i])
3      for j=1 to NDOF do
4          GC=G[j]
5          JVAL=0
6          for k=1 to NN do
7              JVAL=JVAL + DIFF[k]*GC[k]
8          JACOBIAN[i,j]=JVAL
```

Note that GC is assigned the name of a *global variable* (say X), in line 4 and, in line 7, assumes symbolic values of associated nodal coordinates (X(k) for k=1 to NN) simply by appending the subscript, "[k]".

2.2.4 Formulation of the Inverse of the [A] Matrix

The inverse of the [A] matrix of equations (12), (13), (14) and (17) is determined using the following:

Expanding equation (12), for the case of 3D cartesian coordinates produces

$$\begin{Bmatrix} \{Un\} \\ \{Vn\} \\ \{Wn\} \end{Bmatrix}_{3m\times 1} = \begin{bmatrix} \{S1(Xn,Yn,Zn)\}_{m\times m} & 0 & 0 \\ 0 & \{S2(Xn,Yn,Zn)\}_{m\times m} & 0 \\ 0 & 0 & \{S3(Xn,Yn,Zn)\}_{m\times m} \end{bmatrix} \begin{Bmatrix} a \end{Bmatrix}_{3m\times 1}$$

in which 'm' represents the number of nodes and {Un}, {Vn}, {Wn} are the nodal values of the displacements. The vectors {S1}, {S2} and {S3} are the shape functions for each of the degrees of freedom (U, V, W), evaluated at the nodal values. That is, the ith row of {Sj} is the value of the shape function for the jth degree of freedom evaluated at the ith node.

For the case in which each degree of freedom is governed by the same shape function, all {Sj} submatrices are equal. The inverse of the [A] matrix, which is of order (3m × 3m), can be obtained by inverting one of the {Sj} matrices of order (m × m) and assembling N copies (where N is the number of degrees of freedom per node) along the main diagonal of a previously zeroed matrix. In addition, the determinant of {Sj} may be factored from the inverse.

For unique values of the {Sj}, each submatrix would have to be inverted. In either case, the computational savings is appreciable: *Rather than* inverting a matrix of order (Nm × Nm), only K inverses of order (m × m) need be taken where K is equal to unity if all shape functions are the same or N if all are unique.

2.2.5 Processing of Strain Components

MACSYMA's General Pattern Matching Functions are used to convert user specifications of strain components into a database from which the {B} matrix for the isoparametric formulation and the {Ba} matrix for the generalized coordinate method are created.

The procedure consists of associating predicates with pattern variables and defining functions of forms, containing these variables, against which input may be tested.

This process permits components to be specified in a calculus notation. For example, $\partial u/\partial x$ is input as D(u,x) and $\partial^2 u/\partial x \partial y$ may be entered as D(u,x,y). Currently permissible forms for components are

a ☆ D(u,x)
a ☆ D(u,x) + b ☆ D(u,x)
u/a

where a, b represent scalars,
u represents a displacement variable,
x represents a global coordinate variable.

SYSTEM IMPLEMENTATION

The present implementation will parse terms specifying second derivatives but will output a warning that the incorporation of such derivatives in the {B} or {Ba} matrices is beyond present capabilities.

The database is in matrix form where each row contains the information associated with a term in the specified strain component. A single component may have more than one algebraic term. Letting DB[i,j] denote a particular element in the database, the meaning of the entries in the ith row is:

DB[i,1] = the number of the component, which is the same as the row that the term will occupy in the {B} or {Ba} matrix,

DB[i,2] = the scalar multiple,

DB[i,3] = index of the displacement variable with respect to the list of displacement variables,

DB[i,4] = index of the first global coordinate variable in the derivative with respect to the list of coordinate variables, tagged as the number of degrees of freedom plus unity for components of the form, "u/a".

DB[i,5] = index as in column 4 but for the second coordinate variable (if any) in the derivative, set to zero since second derivatives are not yet implemented.

As an example, the database for the strain components,

$$\{ \partial u/\partial r,\ u/r,\ \partial w/\partial z,\ \partial u/\partial z + \partial w/\partial r \}$$

given the list of displacement and coordinate variables as [r, z] and [u, w] respectively is

1	1	1	1	0
2	1/r	1	3	0
3	1	2	2	0
4	1	1	2	0
4	1	2	1	0

The strain components would be entered as

{ D(U,R), U/R, D(W,Z), D(U,Z)+D(W,R) }

The user is given the option of either specifying the strain components using the above notation or by making a selection from a library of pre-stored databases.

The current library options are:

(1) User-Supplied Values

(2) One Dimensional Elasticity
(3) Plane Stress
(4) Plane Strain
(5) Axisymmetric
(6) Linear Isotropic Elasticity - 3D

Options (3) and (4) reference identical values and are listed as two separate options only for conformability with the Material Properties library options (see section 2.7). Option (1) is provided in the event that the user has designated library specification but the provided options are not appropriate.

Examples of both user and library specifications are presented in Chapter 3.

2.2.6 Generation of the {B} Matrices

The synthesis of the {B} or the {Ba} matrix requires the generation of the appropriate derivatives of the shape functions and the assembly of these derivatives according to the specification of the database presented in section 2.2.5.

For the generalized coordinate formulation, the algorithm for the synthesis of the {Ba} matrix may be abstracted as:

Let

NSC = the total number of strain components,
NST = the total number of strain terms,
NN = the number of element nodes,
NDOF = the number of degrees of freedom per node,
S = the vector representing the shape function,
D, FACTOR, ROW, START, and COL are local variables.

then

```
Zero a matrix of order NSC by (NN * NDOF)
for i=1 to NST do
     START = (DB[i,3] - 1) * NN
     if DB[i,4]=NDOF+1
        then D = S
        else D = (the derivative of S with respect to the
                    coordinate variable specified in DB[i,4])
     FACTOR = DB[i,2]
     ROW = DB[i,1]
     for j=1 to NN do
          COL = START + j
          Ba[ROW,COL] = Ba[ROW,COL] + FACTOR*D
```

The algoritm for the generation of the {B} matrix for the isoparametric formulation is presented in two parts: the generation of the derivatives and the assembly process.

SYSTEM IMPLEMENTATION

To generate a matrix, DTEMP, the rows of which contain the derivatives of the shape functions, S, with respect to the global coordinates, perform the following:

```
Let
     NDOF = the number of degrees of freedom per node,
     NN = the number of nodes,
     S = the vector of shape functions,
     D, JVAL are local variables.
Then

1  Generate ADJJ, the adjoint of the Jacobian
2  for j=1 to NDOF do
3       D = (derivative of S with respect to
               the jth natural coordinate)
4       for i=1 to NDOF do
5            JVAL = ADJJ[i,j]
6            for k=1 to NN do
7                 DTEMP[i,k] = DTEMP[i,k] + JVAL*D[k]
8  Append a copy of S (for the 0th derivative) as the last ( i.e.
   NDOF+1st row ) to facilitate processing of terms of the form
   "u/a" .
```

The process of assembling the derivatives to form the {B} matrix is abstracted in the following algorithm which closely resembles its counterpart in the generalized coordinate approach.

```
Let
     NN, NDOF, DB, DTEMP, NSC, NST = as previouly defined
     START1, START2, ROW, FACTOR, ROW, COL are local variables

then

1  Zero a matrix, {B} of order NSC by (NN * NDOF)
2  for i=1 to NST do
3       START1 = (DB[i,3] - 1) * NN
4       START2 = DB[i,4]
5       FACTOR = DB[i,2]
6       ROW = DB[i,1]
7       for j=1 to NN do
8       COL = START1 + j
9       B[ROW,COL] = B[ROW,COL] + FACTOR*DTEMP[START2,j]
```

Whereas the algorithm for the generalized coordinate approach generates the derivatives 'on-the-fly', the derivatives referenced in the algoritm above are prestored in the DTEMP array.

To identify the occurrence of the "u/a" forms, DB[i,4] and hence START2 (line 4) were set to NDOF+1. The algorithm generating DTEMP appends the 0th derivative of the shape function as the NDOF+1st row.

2.2.7 Material Properties Specification

The user has two options in the specification of the Material Properties Matrix of equation (9) (see section 1.2.1):

For the "User-Supplied" option, the user may enter the upper triangular portion of the material properties matrix in row major format. Specification of a constant scalar multiple is also accepted. Editing functions which permit the display or modification of either the matrix or scalar multiple, are provided.

The "Library" option permits access to a set of pre-stored material properties matrices. The library options coincide with those for the strain component specification:

(1) User-Supplied Values
(2) One Dimensional Elasticity
(3) Plane Stress
(4) Plane Strain
(5) Axisymmetric
(6) Linear Isotropic Elasticity - 3D

Option (1) is provided in the event that the user has designated library specification but the provided options are not appropriate.

The library matrices use the symbols, "E" and "NU" to represent Young's Modulus and Poisson's Ratio respectively. Until an edit function, which would permit the renaming of these values, is created, the user should avoid using these names in the specification of other variables.

2.2.8 Element Volume Modification

The nominal elemental volume is calculated as the product of the differential forms of the global coordinates for the generalized coordinate approach or of the natural coordinates for the isoparametric formulation.

The user may modify these forms by inputting appropriate scalar multiples. For example, in axisymmetric problems, the user might enter the factor, $2\pi r$, as a multiple of the products of differentials; $dr{*}dz$. The modification terms, also referred to as Auxiliary Terms, may contain any syntactically correct arithmetic expressions. Examples of the use of element volume modification terms are presented in Chapter 3.

These values are placed on a stack upon input. The system uses the same stack to retain auxiliary terms encountered during computation (for example, the determinant of [A] or of the Jacobian).

2.2.9 Determination of the Limits of Integration

In the generalized coordinate formulation, numeric values for the limits of integration cannot be established until the element is located in the finite element mesh: only then can maximum and minimum coordinate values be determined. To circumvent this problem the limits are taken as identifiers formed by concatenating the coordinate variable names with the strings "MIN" and "MAX" (for example XMIN, XMAX, YMIN and YMAX). The matrix template produced is thus a function of these values. Numerical values for the -MIN and -MAX variables may be established at execution time by applying the FORTRAN intrinsics AMIN1 and AMAX1 to the appropriate nodal coordinate vector.

The determination of the limits of integration for the isoparametric formulation is conditional upon whether the natural coordinates are independent or dependent.

If the coordinates are independent, the user specifies limits from the set: { -1, 0, 1 } with the stipulation that the lower limit is less than the upper limit.

If the coordinates are dependent, the lower limit may be specified but the upper limit is prescribed by the following relationship:

Let N(i) denote the ith natural coordinate. Then

$$\text{upper limit of } N(i) = \begin{cases} 1 & \text{if } i = 1 \\ 1 - \sum_{j=1}^{i-1} N(j) & \text{otherwise} \end{cases}$$

2.2.10 Integration Processing

The steps in performing the integration for both the generalized coordinate and isoparamatric methods, as given in equations (2) and (16) respectively, are comparable:

(a) Form the quadratic form involving the [C] matrix and either {B} or {Ba}.

(b) Multiply the result by any auxiliary terms or element volume modifier terms which are functions of the variables of integation.

(c) Integrate the terms in the upper triangular portion with respect to each coordinate and evaluate at the limits of integration.

2.3 System Description

2.3.1 Logic Hierarchy

***** is divided into program units call 'phases.' Each phase performs a specific computational or system bookkeeping task.

A phase consists of a command file and a function file. The command file is a collection of MACSYMA commands and invocations of system functions and is executed via the MACSYMA 'BATCH' command. Phases may, in turn, execute other phases by issuing the appropriate BATCH command. The function file contains the definition of the system functions invoked in the command file and all of their external references. To minimize the amount of storage, the function file is loaded at the start of the phase and all functions are deleted before the phase is exited. In addition, run configuation processes which perform a trace, dump and break (see section 2.6) are enacted.

The general format of a command file is:

(a) Print the header message giving the decription of the phase,

(b) If the phase is to be traced, turn on the trace facility,

(c) Load the function file,

(d) Execute the system functions,

(e) If specified, perform a dump,

(f) if specified, perform a break,

(g) Remove all functions loaded from the function file.

2.3.1.1 Hierarchy for the Isoparametric Formulation

This section contains a listing of the phases, in the order in which they are executed, for the isoparametric formulation. Several of the phases are common to both formulations.

FSTAR - System Entry Point
The user initiates execution of the system by issuing the command, 'BATCH(IFSTAR,CMD,DSK,AK1G1,ON);'. This phase also opens the disk file which is to contain a record of the execution and loads a set of global system functions. The phase then executes phases SYINIT, SELECT and TERMIN.

SYINIT - System Initialization
Set the operating environment, print opening messages and input the user identification.

SELECT - Method Selection
Input the user specification of the formulation (isoparametric or generalized coordinate) and initiate execution of the appropriate

formulation executive.

ISOEXC - Isoparametric Formulation Executive
Execute phases associated with the isoparametric formulation. This phase requires no user input.

ISOEIN - Isoparametric Formulation Initialization
Process user specifications for the run configuration (TRACE, DUMP and BREAK settings).

INPISO - Problem Parameter Specification
Input and process user specifications of the number of element nodes, number of degrees of freedom per node, number of natural coordinates and the names of the global coordinates, natural coordinates and displacement variables.

SFNISO - Shape Function Processor
Input the shape functions and, if necessary, express them in terms of an independent set of natural coordinates.

BMDATA - B Matrix Database Generation
Process strain component specifications to synthesize the database to be used in the generation of the {B} and {Ba} matrices. Specifications may be either by library selection or manual input.

MATERL - Material Properties Selection
Process specification of material properties by either library selection or manual input.

AUXTER - Auxiliary Term Processor
Input and store the factors modifying the elemental volume.

LIMTIS - Integration Limits
Check if the natural coordinates are dependent or independent. Input and process limits accordingly.

JACOBN - Jacobian Generation
Form the Jacobian matrix and its determinant. This phase requires no input.

BMXISO - {B} Matrix Generation
Form the {B} matrix as specified by the database. No user input is required.

INTISO - Integration Processor
See section 2.2.10 Integration Processing. No user input is required.

DISIPP - Display Pre-processor
Algebraically simplify the results of the integration phase to facilitate evaluation. User input is not required.

DISPLY - Display Processor
Input user specification of the output format of the stiffness

matrix. Process accordingly.

TERMIN - System Termination
Print/closing messages, close the file containing the record of the execution and process the specifications for terminating the run.

2.3.1.2 Hierarchy for the Generalized Coordinate Formulation

This section contains a listing of the phases, in the order in which they are executed, for the generalized coordinate formulation. For phases which are common to both formulations reference is made to the description in the previous section.

FSTAR - System Entry Point
(See previous section)

SYINIT - System Initialization
(See previous section)

SELECT - Method Selection
(See previous section)

GECEXC - Generalized Coordinate Formulation Executive
Execute phases associated with the generalized coordinate formulation. No user input is required.

GECEIN - Generalized Coordinate Formulation Initialization
Process user specifications foe the run configuration (TRACE, DUMP and BREAK settings).

INPGEC - Problem Parameter Specification
Input and process the user specifications for the number of nodes, number of degrees of freedom per node and the names of the displacement and global coordinate variables.

SFNGEC - Shape Function Processor
Input and process shape functions.

BMDATA - {B} Matrix Database Generation
(See previous section)

MATERL - Material Properties Selection
(See previous section)

AUXTER - Auxiliary Term Processor
(See previous section)

LIMTGC - Integration Limits
Determine the limits of integration.
No user input is required.

AMXINV - Inversion of the [A] Matrix
See the algorithm in section 2.2.4. No user input is required.

BMXGEC - (Ba) Matrix Generation
Form the (Ba) matrix as specified by the database. No user input is required.

INTGEC - Integration Processor
See the description in section 2.2.10. No user input is required.

DISGPP - Display Pre-processor
Algebraically simplify the form of the stiffness matrix to facilitate evaluation. No user input is required.

DISPLY - Display Processor
(See the previous section)

TERMIN - System Termination
(See the previous section)

2.3.1.3 Execution Hierarchy

Upon completion of a phase, control returns to the module which issued the BATCH command invoking it. For phases ISOEIN through DISPLY for the isoparametric method, the invoking module is ISOEXC which then executes the BATCH command for the next phase in that sequence. An identical relationship exists between the sequence of phases GECEIN through DISPLY with respect to GECEIN for the generalized coordinate formulation. When execution of these executive phases is completed, control reverts to SELECT. Phase FSTAR executes phases SYINIT, SELECT and TERMIN.

This hierarchy is advantageous in that it facilitates the reorganization of existing phases and the addition of new or alternative ones.

2.3.2 Data Flow

Data Flow refers to the transfer of data values among the different program segments which comprise the system. The designation, 'system value', will refer to those values which represent a major term in the computation and which are not local to the phase in which they are created.

MACSYMA's rules of scope are similar to those implemented in ALGOL 60. Thus, system values are automatically placed in a global pool, the MACSYMA 'VALUES' list, simply by not declaring then local to any function. *Parameters local to a function appear in* brackets at the beginning of the function body as prescribed by MACSYMA syntax.

To increase programming clarity, the following design decisions relating to data flow were made:

(a) System values are returned only through functions explicitly invoked in command files. MACSYMA functions, like LISP functions, can only return a single value. If multiple values are to be generated by the same function, they are returned as elements in a simple list and unpacked in the command file.

(b) Parameters of a function must appear in the argument list. Though the system values are global in scope and thus referencable in the body of any function, this stipulation was made in keeping with good program modularity.

User specified values of element volume modification factors and other auxiliary terms are retained on a simple stack. This stack is also used to retain values resulting from internal computation (for example, the determinant of the Jacobian and of the [A] matrix).

The integration processor examines the stack for expressions which are functions of the variables of integration and multiplies them into the quadratic form (see section 2.2.10). The display processors will identify those elements in the stack which are not functions of these integration variables and collect them into a single expression. This expression constitutes a common factor of the stiffness matrix.

A list of system values, the phase in which they are created (designated in parentheses) and a brief definition follows:

AINVERSE - (AMXINV)
The factored [A] matrix, referred to as [AJ] in section 2.2.1

BMATRIX - (BMXISO, BMXGEC)
The {B} or {Ba} matrix

BMATRIXDATABASE - (BMDATA)
Database containing the specification for the synthesis of the {B} of {Ba} matrices (see section 2.2.6)

BREAKPOINTS - (ISOEIN, GECEIN)
List of the phases for which a BREAK is requested

DELAYSWITCH - (SYSINT)
Batch file processing switch, initialized to 'ON', necessary to overcome the inability of MACLISP, the base language for MACSYMA, to have more than one file open at a time.

DETAMATRIX - (AMXINV)
Determinant of the [A] matrix

DETJACOBIAN - (JACOBN)
Determinant of the Jacobian matrix

DUMPPOINTS - (ISOEIN, GECEIN)
List of the phases for which a DUMP is requested

DISPLACEMENTS - (INPISO, INPGEC)

List of the names of the displacements

FACTOR - (DISIPP, DISGPP)
Common scalar multiple of FUNCTIONAL

FUNCTIONAL - (INTISO, INTGEC)
Matrix of stiffness coefficients with common multiples factored

GLOBALCOORDS - (INPISO, INPGEC)
List of the names of the global (geometry) coordinates

HELP - (SYINIT)
String entered to request the help text, initialized to '?'

JACOBIAN - (JACOBN)
The Jacobian matrix

LIMITS - (LIMTIS, LIMTGC)
Matrix, the rows of which contain the upper and lower limits of integration

MATERIALSMATRIX - (MATERL)
Material properties matrix with common multiple factored

MATERIALSMATRIXMULTIPLIER - (MATERL)
Common multiple of MATERIALSMATRIX

METHOD - (SELECT)
Integer corresponding to the chosen formulation: 1=isoparamatric, 2=generalized coordihate

NATURALCOORDS - (INPISO, INPGEC)
List of the names of the natural coordinates

NDOF - (INPISO, INPGEC)
Number of degrees of freedom per node

NUMNATURAL - (INPISO, INPGEC)
Number of natural coordinates

NUMNODES - (INPISO, INPGEC)
Number of element nodes

NUMSTRAINCOMPONENTS - (BMDATA)
Number of strain components

NUMSTRAINTERMS - (BMDATA)
Total number of strain terms; a strain component may contain more than one algebraic term

PHASENAMES - (ISOEIN, GECEIN)
List of phase names

SHAPEFUNCTIONS - (SFNISO, SFNGEC)
Matrix, the rows of which contain the shape functions

STACK - (SYINIT, AUXTER)
Push-down stack containing the element volume modification factors, and auxiliary terms

STACKPOINTER - (SYINIT)
Pointer to just below the top element in STACK

TRACEPOINTS - (ISOEIN, GECEIN)
List of the phases for which a TRACE is requested

USERNAME - (SYINIT)
User identification; used to name the Record File

2.4 Help Text

Anticipating that diagnostic messages and prompts may not provide sufficient information about a particular input value, a mechanism providing on-line assistance is incorporated.

The user may answer any prompt with "?" and a brief help text will be printed at the terminal. The text contains a description of the requested input and, if a selection is to be made, an enumeration of the alternatives.

Upon exiting the help text, control returns to the prompt for the input.

2.5 Input and Output Functions

2.5.1 Record File

To assist in diagnostics and system development, a record of each execution of the system is made. The record is written to a disk file and contains a copy of all user input, command executions and system dumps.

2.5.2 Input and Output Facilities

In the default mode, MACSYMA commands are echoed at the terminal as they are typed in if entered interactively or as they are executed if prestored on a disk file. A command terminated with a semicolon will display the returned value, while if terminated with a dollar sign, will not. All output to the terminal, excluding the echo of typed input is suppressed if the switch, 'TTYOFF' is set to TRUE. This condition prevails until the switch is reset to FALSE.

As ***** is intended to be an interactive system, it is necessary to selectively suppress output to the user's terminal: the echo of commands and the automatic display of the results retrieved by functions is prevented but prompts, help text, error messages and requested values are displayed. In addition, a complete record of user input, all command executions and values returned by command level functions is placed on the file containing the record of the execution. To accomplish this, the following steps were implemented:

(a) All function calls in the command files are terminated with a dollar sign thus suppressing the automatic display of returned values.

(b) The TTYOFF switch is set to TRUE in phase FSTAR. System output functions locally set TTYOFF to FALSE, execute MACSYMA display commands and then reset TTYOFF. Thus all output is suppressed except that which is channeled through the system output functions.

(c) Since TTYOFF only affects output to the user's terminal, the file containing the record of the execution is unaffected.

2.5.3 Input Characteristics

The use of the MACSYMA 'READ' function presents some difficulties. The function will read in and evaluate one expression. If the input corresponds to the name of a variable local to the function containing the READ or one in the global pool, the evaluation step causes the input expression to be replaced by the value of that variable. The resolution was provided, upon request from MACSYMA system programmers in the form of the heretofore undocumented 'READIN' function. The specification for the READIN function is the same as that for READ with the exception that the input expression is not evaluated.

All input is format free with the exception that expressions must be terminated with either a semicolon or a dollar sign. It is not necessary to press the RETURN key. Either upper or lower case may be used.

2.6 Run Configuration

Run Configuration refers to the state of the TRACE, BREAK and DUMP facilities which may be set for individual phases. These features are intended primarily for system maintenance.

The TRACE facility causes an echoing of the commands in the command file as they are executed.

If the BREAK switch is set for a phase, execution will be suspended just prior to the phase termination and control will revert to MACSYMA command level. Execution resumes upon entry of "EXIT;".

Whereas TRACE and BREAK processes are intended purely for system maintenance, the DUMP feature may be valuable to the user. If DUMP is set, the system values generated during a phase are displayed at the user's console. Independent of this setting, these values are always dumped to the file containing the record of the execution. As certain dumps are quite lenghty, this feature should be used judiciously.

The run configuration may be set in the phases ISOEIN and GECEIN.

2.7 Error Recovery

System error recovery emcompasses three processes:

(1) Semantic errors are trapped during error checking, by system functions. A diagnostic is printed and the user is prompted to re-enter. If the diagnostic offers insufficient explanation, the user may answer the prompt with "?;" which is the request for the help text.

(2) Syntactic errors are trapped by MACSYMA. The erroneous input is echoed along with a pointer indicating the subexpression in error. A diagnostic is printed and the message, "Please rephrase or edit," is displayed. The user may either retype the input or press the escape key to enter the MACSYMA editor. The editor, a derivative of TECO, is documented in Bogen[7].

(3) Errors which cannot be trapped by either of the above procedures are associated with a system failure and are fatal. The system will attempt to shutdown as neatly as possible by preserving the global pool of values and closing the file containing the record of execution so that diagnostics may be run. A message directing the user to sources of assistance is displayed and the user is queried for the system termination procedure as in a normal run.

2.8 Limitations

The limitations of the present system are:

(a) Each node must have the same degrees of freedom;

(b) Each degree of freedom must be represented by the same shape function;

(c) Allowable forms for the specification of strain components are:

a * D(u,x)
a * D(u,x) + b * D(u,x)
u/a

where 'a' and 'b' are scalars, 'u' represents a displacement variable and 'x' represents a geometry (global coordinate) variable.

(d) Problem size is limited by available list space on the computer supporting MACSYMA.

Minor modification will permit the processing of second derivatives andprovisions in the program have been incorporated to facilitate the elimination of restrictions (a) and (b).

2.9 Implementation of Element Loads, Mass and Stresses

The present version of the processor may be extended to permit the implementation of the quantities defined in section 1.2.3.

Many of the constituent matrices are formulated as part of the synthesis of the stiffness matrix. These include the {B} and [C] matrices and the matrix of shape functions. The remaining constituent matrices may be taken from user specification of [m], {fb}, {ft}, {ei} and the names of the nodal displacements.

Integration functions can be used to evaluate the required volume and surface integrals.

Chapter 3

ILLUSTRATIVE EXAMPLES

This chapter presents a number of illustrative examples processed on *****. The bulk of the presentation consists of the execution record. Comments are placed in angle brackets ("<", ">") and were not part of the execution.

The internal storage scheme for the stiffness matrix is a vector, "FUNCTIONAL", containing the elements of the lower triangular portion in row major format and a variable, "FACTOR", containing the common multiple of each element. In the verification sections for each example, these quantities are retrieved and MACSYMA functions are used to recast the results into alternate forms and to establish identities.

MACSYMA functions employed in the verification sections include EV, which evaluates its first argument in the environment specified by the remaning arguments, and FACTOR, which factors its argument into factors irreducible over integers. (The function, FACTOR, should not be confused with the ***** variable, FACTOR. They are readily distinguisable in that the function requires an argument in parenthesis). Details of these and the other functions may be found in reference [7].

The examples are presented as follows:

Section 3.1 - Bar Element with Uniform Cross-Sectional Area
Part 3.1.1 - Generalized Coordinate Formulation
Part 3.1.2 - Isoparametric Formulation
Part 3.1.3 - Comparison and Verification

Section 3.2 - Bar Element with Linear Variation of Cross-Sectional Area
Part 3.2.1 - Generalized Coordinate Formulation
Part 3.2.2 - Isoparametric Formulation
Part 3.2.3 - Comparison and Verification

Section 3.3 - Constant Strain Triangle (CST) with Uniform Thickness
Part 3.3.1 - Generalized Coordinate Formulation
Part 3.3.2 - Isoparametric Formulation
Part 3.3.3 - Comparison and Verification

Section 3.4 - Four Node Quadrilateral
Part 3.4.1 - Generalized Coordinate Formulation
Part 3.4.2 - Calibration
Part 3.4.3 - Isoparametric Formulation

ILLUSTRATIVE EXAMPLES

The final section is devoted to a discussion of the algebraic form of the results.

ILLUSTRATIVE EXAMPLES

3.1 Bar Element with Uniform Cross-Sectional Area

<The bar element has one degree of freedom, in the axial direction, and is modeled using a linear displacement function.>

3.1.1 Generalized Coordinate Formulation

<Immediately subsequent to the LOGIN procedure, a copy of MACSYMA is loaded (via ":A") and the BATCH command is issued to initiate execution of *****.>

```
*:A
This is MACSYMA 266
FIX266 8 DSK MACSYMA being loaded
loading done

(C1) BATCH([FSTAR,CMD,DSK,AK1G],ON);

(C2) TTYOFF:TRUE$

 * SYSTEM INITIALIZATION *

WELCOME TO ***** VERSION 1.0

It is now   THURSDAY  DECEMBER 1,1977  18:2:10
The current file is [SYINIT, FCN]
The current device and username is [DSK, AK1G]

Report problems to
ALAN R. KORNCOFF
DEPT. OF CIVIL ENGINEERING
CARNEGIE-MELLON UNIVERSITY
CMU-10A, AK1G

Terminate all input with a SEMICOLON - ';'
Input '?;' for HELP


Input your LOGIN NAME
?
AK1G;
Is AK1G correct ?  (YES; or NO;)
?
YES;

GREETINGS AK1G

 * METHOD SELECTION *

The available formulations include
   (1) THE ISOPARAMETRIC METHOD
   (2) THE GENERALIZED COORDINATE METHOD
```

```
Please enter the number of the method chosen (1 OR 2).
?
2;
GENERALIZED COORDINATE FORMULATION

 * GENERALIZED COORDINATE FORMULATION EXECUTIVE *

 * GENERALIZED COORDINATE FORMULATION INITIALIZATION *

Do you wish to set DUMP, BREAK or TRACE POINTS
?
YES;

1. GECEIN       2. INPGEC       3. SFNGEC       4. BMDATA
5. MATERL       6. AUXTER       7. LIMTGC       8. AMXINV
9. BMXGEC       10. INTGEC      11. DISGPP      12. DISPLY
SET DUMP POINTS
Designate selected PHASES by entering
the associated integer INDEX or 'ALL' for all phases.
Type 'END' to TERMINATE.
?
2;
?
4;
?
5;
?
END;

1. GECEIN       2. INPGEC       3. SFNGEC       4. BMDATA
5. MATERL       6. AUXTER       7. LIMTGC       8. AMXINV
9. BMXGEC       10. INTGEC      11. DISGPP      12. DISPLY
SET BREAKPOINTS
Designate selected PHASES by entering
the associated integer INDEX or 'ALL' for all phases.
Type 'END' to TERMINATE.
?
END;

1. GECEIN       2. INPGEC       3. SFNGEC       4. BMDATA
5. MATERL       6. AUXTER       7. LIMTGC       8. AMXINV
9. BMXGEC       10. INTGEC      11. DISGPP      12. DISPLY
SET TRACE POINTS
Designate selected PHASES by entering
the associated integer INDEX or 'ALL' for all phases.
Type 'END' to TERMINATE.
?
END;

 * PROBLEM PARAMETER SPECIFICATION - GENERALIZED COORDINATE *

Input the NUMBER OF ELEMENT NODES
?
```

```
2;

Input the NUMBER OF DEGREES OF FREEDOM PER NODE
?
1;

Input the vector of the NAMES OF THE GLOBAL COORDINATES
There should be 1 elements
ELEMENT 1 =
X;

Input the vector of the NAMES OF THE DISPLACEMENT VARIABLES
There should be 1 elements
ELEMENT 1 =
U;

DUMP FOR INPGEC
                                   NUMNODES = 2

                                   NDOF = 1

                                   GLOBALCOORDS = [X]

                                   DISPLACEMENTS = [U]


 * SHAPE FUNCTION PROCESSOR - GENERALIZED COORDINATE *

ENTER the terms of the SHAPE FUNCTION ordered from
GENERALIZED COORDINATE 1 through coordinate 2.
ELEMENT 1 =
1;
ELEMENT 2 =
X;

< This input represents the displacement  function: u = a1 + a2*x , in
which "a1" and "a2" are the generalized coordinates.>

SHAPE FUNCTION MODIFICATION
The OPTIONS are
   (1) DISPLAY THE SHAPE FUNCTIONS
   (2) MODIFY THE SHAPE FUNCTIONS
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
1;

The terms of the SHAPE FUNCTION are
Term 1
                                   1

Term 2
```

```
                              X

Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;

 * B MATRIX DATABASE GENERATION *

The OPTIONS for specifying STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) LIBRARY VALUES
Please ENTER the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) ONE DIMENSIONAL ELASTICITY
   (3) PLANE STRESS
   (4) PLANE STRAIN
   (5) AXISYMMETRIC
   (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
2;

< The library contains pre-stored strain components specifications in
the database format described in section 2.2.5.>

DUMP FOR BMDATA
                        BMATRIXDATABASE = [ 1  1  1  1  0 ]

                                  NUMSTRAINTERMS = 1

                               NUMSTRAINCOMPONENTS = 1


 * MATERIAL PROPERTIES SELECTION *

The options for the selection of the MATERIAL PROPERTIES MATRIX are
   (1) USER-SUPPLIED MATRIX
   (2) LIBRARY MATRIX
Please enter the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING MATERIAL PROPERTIES are
   (1) USER-SUPPLIED VALUES
   (2) ONE DIMENSIONAL ELASTICITY
   (3) PLANE STRESS
   (4) PLANE STRAIN
   (5) AXISYMMETRIC
   (6) LINEAR ISOTROPIC ELASTICITY - 3D
```

ILLUSTRATIVE EXAMPLES

Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
2;

DUMP FOR MATERL

MATERIALSMATRIX = [1]
MATERIALSMATRIXMULTIPLIER = E

☆ AUXILIARY TERM PROCESSOR ☆

ENTER ELEMENT VOLUME MODIFICATION FACTORS OR AUXILIARY TERMS
TYPE 'END;' to TERMINATE
?
A;
?
END;

AUXILIARY TERM MODIFICATION
The OPTIONS are
(1) DISPLAY THE AUXILIARY TERMS
(2) MODIFY AN AUXILIARY TERM
(3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;

☆ INTEGRATION LIMITS - GENERALIZED COORDINATE ☆

☆ INVERSION OF MATRIX, A ☆

☆ B MATRIX GENERATION - GENERALIZED COORDINATE ☆

☆ INTEGRATION PROCESSOR - GENERALIZED COORDINATE ☆

☆ DISPLAY PRE-PROCESSOR - GENERALIZED COORDINATE ☆

☆ DISPLAY PROCESSOR ☆

The options for OUTPUTTING the STIFFNESS MATRIX are
(1) Upper triangular portion in algebraic format
(2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED with YOUR SELECTION.
?
2;

Enter the FORTRAN ARRAY NAME for the STIFFNESS MATRIX
?
STIFF;

ILLUSTRATIVE EXAMPLES

```
Multiply each coefficient by

     A*E/(X(2)-X(1))**2

The STIFFNESS MATRIX is

     STIFF(1,1) = -(XMIN-XMAX)
     STIFF(2,1) = XMIN-XMAX
     STIFF(3,1) = -(XMIN-XMAX)


 * SYSTEM TERMINATION *

That's all for now, AK1G
It is now 12/1/77   18:13:57

Accumulated CPU TIME = 18348 MSEC

A record of this session is recorded in file [AK1G, >, DSK, USERS ]

The options for exiting the system are
   (1) TERMINATE THE RUN
   (2) TERMINATE THE RUN AND THE JOB
   (3) TERMINATE THE RUN, THE JOB AND LOGOUT

Please enter the number of the option chosen (1, 2 or 3)
?
3;
Accumulated CPU TIME = 18642 MSEC
TERMINATE AND LOGOUT
:LOGOUT

<The system automatically issued the ":LOGOUT" command. Details of
the termination procedure are given in the Appendix.>

<Option 1 of the DISPLAY PROCESSOR would provide the following
output:>


The options for OUTPUTTING the STIFFNESS MATRIX are
   (1) Upper triangular portion in algebraic format
   (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED with YOUR SELECTION.
?
1;

Multiply each coefficient by
                      A E
                   ----------
                            2
                   (X  - X )
                     2    1
```

ILLUSTRATIVE EXAMPLES

```
ROW 1, COL 1

                    - (XMIN - XMAX)

ROW 2, COL 1

                      XMIN - XMAX

ROW 2, COL 2

                    - (XMIN - XMAX)
```

3.1.2 Isoparametric Formulation

```
 * SYSTEM INITIALIZATION *

WELCOME TO ***** VERSION 1.0

It is now  THURSDAY  DECEMBER 1,1977  1:41:26
The current file is [SYINIT, FCN]
The current device and username is [DSK, AK1G]

Report problems to
ALAN R. KORNCOFF
DEPT. OF CIVIL ENGINEERING
CARNEGIE-MELLON UNIVERSITY
CMU-10A, AK1G

Terminate all input with a SEMICOLON - ';'
Input '?;' for HELP

Input your LOGIN NAME
?
AK1G;
Is AK1G correct ?  (YES; or NO;)
?
YES;

GREETINGS AK1G

 * METHOD SELECTION *

The available formulations include
   (1) THE ISOPARAMETRIC METHOD
```

ILLUSTRATIVE EXAMPLES

(2) THE GENERALIZED COORDINATE METHOD

```
Please enter the number of the method chosen (1 OR 2).
?
1;
ISOPARAMETRIC FORMULATION

 * ISOPARAMETRIC FORMULATION EXECUTIVE *

 * ISOPARARMETRIC FORMULATION INITIALIZATION *

Do you wish to set DUMP, BREAK or TRACE POINTS
?
NO;

 * PROBLEM PARAMETER SPECIFICATION - ISOPARAMETRIC *

Input the NUMBER OF ELEMENT NODES
?
2;

Input the NUMBER OF DEGREES OF FREEDOM PER NODE
?
1;

Input the NUMBER OF NATURAL COORDINATES
?
1;

Input the vector of the NAMES OF THE NATURAL COORDINATES
There should be 1 elements
ELEMENT 1 =
L;

Input the vector of the NAMES OF THE GLOBAL COORDINATES
There should be 1 elements
ELEMENT 1 =
X;

Input the vector of the NAMES OF THE DISPLACEMENT VARIABLES
There should be 1 elements
ELEMENT 1 =
U;

 * SHAPE FUNCTION PROCESSOR - ISOPARAMETRIC *

ENTER the terms of the SHAPE FUNCTION ordered from
node 1 through node 2.
The 2 elements will be prompted for
ELEMENT 1 =
.5*(1-L);
ELEMENT 2 =
.5*(1+L);
```

ILLUSTRATIVE EXAMPLES

```
< This input represents the displacement equation:
 u = 0.5*(1-l) + 0.5*(1+l) .>

SHAPE FUNCTION MODIFICATION
The OPTIONS are
   (1) DISPLAY THE SHAPE FUNCTIONS
   (2) MODIFY THE SHAPE FUNCTIONS
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
1;

The terms of the SHAPE FUNCTION are
Term 1
                                    0.5*(1 - L)

Term 2
                                    0.5*(L + 1)

Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;

 * B MATRIX DATABASE GENERATION *

The OPTIONS for specifying STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) LIBRARY VALUES
Please ENTER the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) ONE DIMENSIONAL ELASTICITY
   (3) PLANE STRESS
   (4) PLANE STRAIN
   (5) AXISYMMETRIC
   (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
2;

 * MATERIAL PROPERTIES SELECTION *

The options for the selection of the MATERIAL PROPERTIES MATRIX are
   (1) USER-SUPPLIED MATRIX
   (2) LIBRARY MATRIX
Please enter the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;
```

ILLUSTRATIVE EXAMPLES

```
The LIBRARY OPTIONS for SPECIFYING MATERIAL PROPERTIES are
   (1) USER-SUPPLIED VALUES
   (2) ONE DIMENSIONAL ELASTICITY
   (3) PLANE STRESS
   (4) PLANE STRAIN
   (5) AXISYMMETRIC
   (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
2;


 * AUXILIARY TERM PROCESSOR *

ENTER ELEMENT VOLUME MODIFICATION FACTORS OR AUXILIARY TERMS
TYPE 'END;' to TERMINATE
?
A;
?
END;

AUXILIARY TERM MODIFICATION
The OPTIONS are
   (1) DISPLAY THE AUXILIARY TERMS
   (2) MODIFY AN AUXILIARY TERM
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;


 * INTEGRATION LIMITS - ISOPARAMETRIC *

ENTER the LIMITS OF INTEGRATION for NATURAL COORDINATE, L
LOWER LIMIT =
?
-1;
UPPER LIMIT =
?
1;

 * JACOBIAN GENERATION - ISOPARAMETRIC *

 * B MATRIX GENERATION - ISOPARAMETRIC *

 * INTEGRATION PROCESSING - ISOPARAMETRIC *

 * DISPLAY PROCESSOR *

The options for OUTPUTTING the STIFFNESS MATRIX are
   (1) Upper triangular portion in algebraic format
   (2) FORTRAN CARD IMAGE format
```

ILLUSTRATIVE EXAMPLES

```
ENTER the NUMBER ASSOCIATED  with YOUR SELECTION.
?
1;

Multiply each coefficient by
                                             2*A*E
                                            -------
                                            X  - X
                                            2    1


ROW 1, COL 1

                                              1
                                              -
                                              2

ROW 2, COL 1

                                               1
                                             - -
                                               2

ROW 2, COL 2

                                              1
                                              -
                                              2


 * SYSTEM TERMINATION *

That's all for now, AK1G
It is now 12/1/77    1:45:24

Accumulated CPU TIME = 17040 MSEC

A record of this session is recorded in file [AK1G, >, DSK, USERS ]

<Display option 2 produces:>


The options for OUTPUTTING the STIFFNESS MATRIX are
    (1) Upper triangular portion in algebraic format
    (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED  with YOUR SELECTION.
?
2;

Enter the FORTRAN ARRAY NAME for the STIFFNESS MATRIX
?
STIFF;
```

ILLUSTRATIVE EXAMPLES

Multiply each coefficient by

/2*A*E/(X(2)-X(1))

The STIFFNESS MATRIX is

```
STIFF(1,1) = 1/2
STIFF(2,1) = -1/2
STIFF(3,1) = 1/2
```

3.1.3 Comparison and Verification

<The files, (%1G,vals) and (%1I,vals) contain the values generated during the runs in sections 3.1.1 and 3.1.2 respectively. For the purposes of comparison, for the generalized coordinate formulation, the coordinate, "X", is taken to be a minimum at node 1 and a maximum at node 2 (i.e. x(1)<x(2)). >

```
time= 4 msec.
(D1)                                [DSK, AK1G]

(C2) LOADFILE(%1I,VALS);

%1I VALS DSK AK1G being loaded
loading done
time= 291 msec.
(D2)                                    DONE

(C3) FUNCTIONAL;
time= 0 msec.
                                      [ 1 ]
                                      [ - ]
                                      [ 2 ]
                                      [   ]
                                      [  1 ]
(D3)                                  [- - ]
                                      [  2 ]
                                      [   ]
                                      [ 1 ]
                                      [ - ]
                                      [ 2 ]

(C4) FTL:FUNCTIONAL*FACTOR;
time= 15 msec.
                               [     A E       ]
                               [   -------     ]
                               [   X  - X      ]
                               [    2    1     ]
                               [               ]
                               [     2 A E     ]
```

ILLUSTRATIVE EXAMPLES

```
(D4)                        [ - ----------- ]
                            [   2 (X  - X ) ]
                            [       2    1  ]
                            [               ]
                            [      A E      ]
                            [    -------    ]
                            [     X  - X    ]
                            [      2    1   ]

(C5) FACTOR(%);
time= 78 msec.
                              [    A E    ]
                              [  -------  ]
                              [  X  - X   ]
                              [   2    1  ]
                              [           ]
                              [    A E    ]
(D5)                          [ -------   ]
                              [  X  - X   ]
                              [   2    1  ]
                              [           ]
                              [    A E    ]
                              [ -------   ]
                              [  X  - X   ]
                              [   2    1  ]
```

< This is the lower triangular portion of the stiffness matrix for the isoparametric formulation expressed in row major format. >

```
(C6) LOADFILE(%1G,VALS);

%1G VALS DSK AK1G being loaded
loading done
time= 258 msec.
(D6)                              DONE

(C7) FTL2:FUNCTIONAL*FACTOR;
time= 15 msec.
                         [   A E (XMIN - XMAX) ]
                         [ - ----------------- ]
                         [            2        ]
                         [       (X  - X )     ]
                         [         2    1      ]
                         [                     ]
                         [  A E (XMIN - XMAX)  ]
                         [  -----------------  ]
(D7)                     [          2          ]
                         [     (X  - X )       ]
                         [       2    1        ]
                         [                     ]
                         [   A E (XMIN - XMAX) ]
                         [ - ----------------- ]
                         [            2        ]
```

```
                    [        (X  - X )       ]
                    [          2    1        ]
```

< The/ XMIN and XMAX terms of the generalized coordinate stiffness matrix in expression (D7) are evaluated at X(1) and X(2) respectively and the vector is simplified. >

```
(C8) FACTOR(EV(%,XMIN=X(1),XMAX=X(2)));
time= 129 msec.
                              [    A E     ]
                              [  -------   ]
                              [  X  - X    ]
                              [   2    1   ]
                              [    A E     ]
(D8)                          [ - -------  ]
                              [  X  - X    ]
                              [   2    1   ]
                              [            ]
                              [    A E     ]
                              [  -------   ]
                              [  X  - X    ]
                              [   2    1   ]
```

< This is the lower triangular portion of the stiffness matix for the generalized coordinate formulation in row major format, evaluated at the boundary conditions. It is identical to the expression in (D5) for the isoparametric formulation. >

```
(C9) CLOSEFILE(%VER,C1);
```

3.2 Bar Element with Linear Variation of Cross-Sectional Area

3.2.1 Generalized Coordinate Formulation

< This run was identical to that of section 3.1.1 with the exception that the cross-sectional area of the bar varies linearly from a value of A(1) at node 1 to a value of A(2) at node 2. The input unique to this execution is shown below. >

```
 * AUXILIARY TERM PROCESSOR *

ENTER ELEMENT VOLUME MODIFICATION FACTORS OR AUXILIARY TERMS
TYPE 'END;' to TERMINATE
?
(X(2)-X)*A(1)/(X(2)-X(1)) + (X-X(1))*A(2)/(X(2)-X(1));
?
END;

AUXILIARY TERM MODIFICATION
The OPTIONS are
   (1) DISPLAY THE AUXILIARY TERMS
   (2) MODIFY AN AUXILIARY TERM
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
1;

The AUXILIARY TERMS are
Term 1
                      A(2)*(X - X(1))   A(1)*(X(2) - X)
                      --------------- + ---------------
                        X(2) - X(1)       X(2) - X(1)

Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;
```

< Both options of the display are: >

```
The options for OUTPUTTING the STIFFNESS MATRIX are
    (1) Upper triangular portion in algebraic format
    (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED  with YOUR SELECTION.
?
1;

Multiply each coefficient by
                                    E
                                ----------
```

ILLUSTRATIVE EXAMPLES

$$(X_2 - X_1)^2$$

ROW 1, COL 1

$$- (XMIN - XMAX)\ (A(2)\ XMIN - A(1)\ XMIN + A(2)\ XMAX - A(1)\ XMAX + 2\ A(1)\ X_2 - 2\ X_1\ A(2))/(2\ (X_2 - X_1))$$

ROW 2, COL 1

$$(XMIN - XMAX)\ (A(2)\ XMIN - A(1)\ XMIN + A(2)\ XMAX - A(1)\ XMAX + 2\ A(1)\ X_2 - 2\ X_1\ A(2))/(2\ (X_2 - X_1))$$

ROW 2, COL 2

$$- (XMIN - XMAX)\ (A(2)\ XMIN - A(1)\ XMIN + A(2)\ XMAX - A(1)\ XMAX + 2\ A(1)\ X_2 - 2\ X_1\ A(2))/(2\ (X_2 - X_1))$$

```
The options for OUTPUTTING the STIFFNESS MATRIX are
   (1) Upper triangular portion in algebraic format
   (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED with YOUR SELECTION.
?
2;

Enter the FORTRAN ARRAY NAME for the STIFFNESS MATRIX
?
STIFF;

Multiply each coefficient by

      E/(X(2)-X(1))**2

The STIFFNESS MATRIX is

      STIFF(1,1) = -(XMIN-XMAX)*(A(2)*XMIN-A(1)*XMIN+A(2)*XMAX-A(1)*XMAX
     1  +2*A(1)*X(2)-2*X(1)*A(2))/(2*(X(2)-X(1)))
      STIFF(2,1) = (XMIN-XMAX)*(A(2)*XMIN-A(1)*XMIN+A(2)*XMAX-A(1)*XMAX+
     1  2*A(1)*X(2)-2*X(1)*A(2))/(2*(X(2)-X(1)))
      STIFF(3,1) = -(XMIN-XMAX)*(A(2)*XMIN-A(1)*XMIN+A(2)*XMAX-A(1)*XMAX
```

ILLUSTRATIVE EXAMPLES

```
1    +2*A(1)*X(2)-2*X(1)*A(2))/(2*(X(2)-X(1)))
```

3.2.2 Isoparametric Formulation

< The input for this case is the same as that for the generalized coordinate formulation. The display options are: >

```
The options for OUTPUTTING the STIFFNESS MATRIX are
   (1) Upper triangular portion in algebraic format
   (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED with YOUR SELECTION.
?
1;
```

Multiply each coefficient by

$$\frac{2E}{X_2 - X_1}$$

ROW 1, COL 1

$$\frac{A(2) + A(1)}{4}$$

ROW 2, COL 1

$$-\frac{A(2) + A(1)}{4}$$

ROW 2, COL 2

$$\frac{A(2) + A(1)}{4}$$

```
The options for OUTPUTTING the STIFFNESS MATRIX are
   (1) Upper triangular portion in algebraic format
   (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED with YOUR SELECTION.
?
2;
```

ILLUSTRATIVE EXAMPLES

```
                                  2
                         (X  - X )
                           2    1



ROW 1, COL 1

- (XMIN - XMAX) (A(2) XMIN - A(1) XMIN + A(2) XMAX - A(1) XMAX

                          + 2 A(1) X  - 2 X  A(2))/(2 (X  - X ))
                                    2      1             2    1

ROW 2, COL 1

(XMIN - XMAX) (A(2) XMIN - A(1) XMIN + A(2) XMAX - A(1) XMAX

                          + 2 A(1) X  - 2 X  A(2))/(2 (X  - X ))
                                    2      1             2    1

ROW 2, COL 2

- (XMIN - XMAX) (A(2) XMIN - A(1) XMIN + A(2) XMAX - A(1) XMAX

                          + 2 A(1) X  - 2 X  A(2))/(2 (X  - X ))
                                    2      1             2    1


The options for OUTPUTTING the STIFFNESS MATRIX are
    (1) Upper triangular portion in algebraic format
    (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED  with YOUR SELECTION.
?
2;

Enter the FORTRAN ARRAY NAME for the STIFFNESS MATRIX
?
STIFF;

Multiply each coefficient by

      E/(X(2)-X(1))**2

The STIFFNESS MATRIX is

      STIFF(1,1) = -(XMIN-XMAX)*(A(2)*XMIN-A(1)*XMIN+A(2)*XMAX-A(1)*XMAX
     1   +2*A(1)*X(2)-2*X(1)*A(2))/(2*(X(2)-X(1)))
      STIFF(2,1) = (XMIN-XMAX)*(A(2)*XMIN-A(1)*XMIN+A(2)*XMAX-A(1)*XMAX+
     1   2*A(1)*X(2)-2*X(1)*A(2))/(2*(X(2)-X(1)))
      STIFF(3,1) = -(XMIN-XMAX)*(A(2)*XMIN-A(1)*XMIN+A(2)*XMAX-A(1)*XMAX
```

```
Enter the FORTRAN ARRAY NAME for the STIFFNESS MATRIX
?
STIFF;

Multiply each coefficient by
      2*E/(X(2)-X(1))

The STIFFNESS MATRIX is

      STIFF(1,1) = (A(2)+A(1))/4
      STIFF(2,1) = -(A(2)+A(1))/4
      STIFF(3,1) = (A(2)+A(1))/4
```

3.2.3 Comparison and Verification

< The files, (%2G,vals) and (%2I,vals) contain values generated during the runs in sections 3.2.1 and 3.2.2 respectively. As in the verification of section 3.1.3, the minimum and maximum values of X are at nodes 1 and 2 respectively. >

```
time= 8 msec.
(D1)                              [DSK, AK1G]

(C2) LOADFILE(%2I,VALS);

%2I VALS DSK AK1G being loaded
loading done
time= 308 msec.
(D2)                                DONE

(C3) FTL:FUNCTIONAL*FACTOR;
time= 14 msec.
                         [  (A(2) + A(1)) E  ]
                         [ ----------------- ]
                         [    2 (X  - X )    ]
                         [        2    1     ]
                         [                   ]
                         [   (A(2) + A(1)) E ]
(D3)                     [ - --------------- ]
                         [     2 (X  - X )   ]
                         [         2    1    ]
                         [                   ]
                         [  (A(2) + A(1)) E  ]
                         [ ----------------- ]
                         [    2 (X  - X )    ]
                         [        2    1     ]

(C4) LOADFILE(%2G,VALS);
```

ILLUSTRATIVE EXAMPLES

```
%2G VALS DSK AK1G being loaded
loading done
time= 377 msec.
(D4)                              DONE

(C5) FTL2:FUNCTIONAL*FACTOR;
time= 18 msec.
(D5) MATRIX([- E (XMIN - XMAX) (A(2) XMIN - A(1) XMIN + A(2) XMAX

                                                     3
 - A(1) XMAX + 2 A(1) X  - 2 X  A(2))/(2 (X  - X ) )],
                       2     1            2    1

[E (XMIN - XMAX) (A(2) XMIN - A(1) XMIN + A(2) XMAX - A(1) XMAX

                                         3
 + 2 A(1) X  - 2 X  A(2))/(2 (X  - X ) )],
           2     1            2    1

[- E (XMIN - XMAX) (A(2) XMIN - A(1) XMIN + A(2) XMAX - A(1) XMAX

                                         3
 + 2 A(1) X  - 2 X  A(2))/(2 (X  - X ) )])
           2     1            2    1

(C6) FACTOR(EV(%,XMIN=X[1],XMAX=X[2]));
time= 711 msec.
                        [  (A(2) + A(1)) E  ]
                        [  ---------------  ]
                        [    2 (X  - X )    ]
                        [        2    1     ]
                        [                   ]
                        [   (A(2) + A(1)) E ]
(D6)                    [ - --------------- ]
                        [    2 (X  - X )    ]
                        [        2    1     ]
                        [                   ]
                        [  (A(2) + A(1)) E  ]
                        [  ---------------  ]
                        [    2 (X  - X )    ]
                        [        2    1     ]
```

< Note that expressions (D6) and (D7) are identical. >

```
(C7) CLOSEFILE(%VER,C2);
```

3.3 CST with Uniform Thickness

< The constant strain triangle (CST) has two degreees of freedom at each node and is modeled with a linear displacement function. >

3.3.1 Generalized Coordinate Formulation

```
 * SYSTEM INITIALIZATION *

WELCOME TO ***** VERSION 1.0

It is now  THURSDAY DECEMBER 1,1977 2:35:9
The current file is [SYINIT, FCN]
The current device and username is [DSK, AK1G]

Report problems to
ALAN R. KORNCOFF
DEPT. OF CIVIL ENGINEERING
CARNEGIE-MELLON UNIVERSITY
CMU-10A, AK1G

Terminate all input with a SEMICOLON - ';'
Input '?;' for HELP


Input your LOGIN NAME
?
AK1G;
Is AK1G correct ?  (YES; or NO;)
?
YES;

GREETINGS AK1G

 * METHOD SELECTION *

The available formulations include
    (1) THE ISOPARAMETRIC METHOD
    (2) THE GENERALIZED COORDINATE METHOD

Please enter the number of the method chosen (1 OR 2).
?
2;
GENERALIZED COORDINATE FORMULATION

 * GENERALIZED COORDINATE FORMULATION EXECUTIVE *

 * GENERALIZED COORDINATE FORMULATION INITIALIZATION *

Do you wish to set DUMP, BREAK or TRACE POINTS
?
```

```
NO;

 * PROBLEM PARAMETER SPECIFICATION - GENERALIZED COORDINATE *

Input the NUMBER OF ELEMENT NODES
?
3;

Input the NUMBER OF DEGREES OF FREEDOM PER NODE
?
2;

Input the vector of the NAMES OF THE GLOBAL COORDINATES
There should be 2 elements
ELEMENT 1 =
X;
ELEMENT 2 =
Y;

Input the vector of the NAMES OF THE DISPLACEMENT VARIABLES
There should be 2 elements
ELEMENT 1 =
U;
ELEMENT 2 =
V;

 * SHAPE FUNCTION PROCESSOR - GENERALIZED COORDINATE *

ENTER the terms of the SHAPE FUNCTION ordered from
GENERALIZED COORDINATE 1 through coordinate 3.
ELEMENT 1 =
1;
ELEMENT 2 =
X;
ELEMENT 3 =
Y;

SHAPE FUNCTION MODIFICATION
The OPTIONS are
   (1) DISPLAY THE SHAPE FUNCTIONS
   (2) MODIFY THE SHAPE FUNCTIONS
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;

 * B MATRIX DATABASE GENERATION *

The OPTIONS for specifying STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) LIBRARY VALUES
```

```
Please ENTER the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING STRAIN COMPONENTS are
  (1) USER-SUPPLIED VALUES
  (2) ONE DIMENSIONAL ELASTICITY
  (3) PLANE STRESS
  (4) PLANE STRAIN
  (5) AXISYMMETRIC
  (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
3;

 * MATERIAL PROPERTIES SELECTION *

The options for the selection of the MATERIAL PROPERTIES MATRIX are
  (1) USER-SUPPLIED MATRIX
  (2) LIBRARY MATRIX
Please enter the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING MATERIAL PROPERTIES are
  (1) USER-SUPPLIED VALUES
  (2) ONE DIMENSIONAL ELASTICITY
  (3) PLANE STRESS
  (4) PLANE STRAIN
  (5) AXISYMMETRIC
  (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
3;

 * AUXILIARY TERM PROCESSOR *

ENTER ELEMENT VOLUME.MODIFICATION FACTORS OR AUXILIARY TERMS
TYPE 'END;' to TERMINATE
?
T;
?
.5;
?
END;
```

< As indicated in section 2.2.7, the nominal element volume is calculated as the product of the differentials of the coordinate variables (in this case as dx*dy). Since the area of the triangle is half this quantity, a volume modification term, equal to .5 is specified. The term, "T", represents the uniform thickness of the element. >

ILLUSTRATIVE EXAMPLES

AUXILIARY TERM MODIFICATION
The OPTIONS are
(1) DISPLAY THE AUXILIARY TERMS
(2) MODIFY AN AUXILIARY TERM
(3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;

* INTEGRATION LIMITS - GENERALIZED COORDINATE *

* INVERSION OF MATRIX, A *

* B MATRIX GENERATION - GENERALIZED COORDINATE *

* INTEGRATION PROCESSOR - GENERALIZED COORDINATE *

* DISPLAY PRE-PROCESSOR - GENERALIZED COORDINATE *

* DISPLAY PROCESSOR *

The options for OUTPUTTING the STIFFNESS MATRIX are
(1) Upper triangular portion in algebraic format
(2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED with YOUR SELECTION.
?
1;

Multiply each coefficient by

$$- E\ T/(2\ (X_2 Y_3 - X_1 Y_3 - Y_2 X_3 + Y_1 X_3 + X_1 Y_2 - Y_1 X_2)^2\ (NU - 1)\ (NU + 1))$$

ROW 1, COL 1

$$- (X_3^2 NU - 2 X_2 X_3 NU + X_2^2 NU - 2 Y_3^2 + 4 Y_2 Y_3 - X_3^2 + 2 X_2 X_3 - 2 Y_2^2 - X_2^2)\ (XMIN - XMAX)\ (YMIN - YMAX)/2$$

ROW 2, COL 1

$$(X_3^2 NU - X_2 X_3 NU - X_1 X_3 NU + X_1 X_2 NU - 2 Y_3^2 + 2 Y_2 Y_3 + 2 Y_1 Y_3$$

$$- X_3^2 + X_2 X_3 + X_1 X_3 - 2 Y_1 Y_2 - X_1 X_2) (\mathrm{XMIN} - \mathrm{XMAX}) (\mathrm{YMIN} - \mathrm{YMAX})/2$$

ROW 2, COL 2

$$- (X_3^2 \,\mathrm{NU} - 2 X_1 X_3 \,\mathrm{NU} + X_1^2 \,\mathrm{NU} - 2 Y_3^2 + 4 Y_1 Y_3 - X_3^2 + 2 X_1 X_3 - 2 Y_1^2$$

$$- X_1^2) (\mathrm{XMIN} - \mathrm{XMAX}) (\mathrm{YMIN} - \mathrm{YMAX})/2$$

ROW 3, COL 1

$$- (X_2 X_3 \,\mathrm{NU} - X_1 X_3 \,\mathrm{NU} - X_2^2 \,\mathrm{NU} + X_1 X_2 \,\mathrm{NU} - 2 Y_2 Y_3 + 2 Y_1 Y_3 - X_2 X_3$$

$$+ X_1 X_3 + 2 Y_2^2 - 2 Y_1 Y_2 + X_2^2 - X_1 X_2) (\mathrm{XMIN} - \mathrm{XMAX}) (\mathrm{YMIN} - \mathrm{YMAX})/2$$

ROW 3, COL 2

$$(X_2 X_3 \,\mathrm{NU} - X_1 X_3 \,\mathrm{NU} - X_1 X_2 \,\mathrm{NU} + X_1^2 \,\mathrm{NU} - 2 Y_2 Y_3 + 2 Y_1 Y_3 - X_2 X_3$$

$$+ X_1 X_3 + 2 Y_1 Y_2 + X_1 X_2 - 2 Y_1^2 - X_1^2) (\mathrm{XMIN} - \mathrm{XMAX}) (\mathrm{YMIN} - \mathrm{YMAX})/2$$

ROW 3, COL 3

$$- (X_2^2 \,\mathrm{NU} - 2 X_1 X_2 \,\mathrm{NU} + X_1^2 \,\mathrm{NU} - 2 Y_2^2 + 4 Y_1 Y_2 - X_2^2 + 2 X_1 X_2 - 2 Y_1^2$$

$$- X_1^2) (\mathrm{XMIN} - \mathrm{XMAX}) (\mathrm{YMIN} - \mathrm{YMAX})/2$$

ROW 4, COL 1

$$(X - X) (Y - Y) (\mathrm{NU} + 1) (\mathrm{XMIN} - \mathrm{XMAX}) (\mathrm{YMIN} - \mathrm{YMAX})$$

```
           3    2    3    2
 - ----------------------------------------------------------
                                 2
```

ROW 4, COL 2

$$(X_3 Y_3 NU - 2 X_2 Y_3 NU + X_1 Y_3 NU + Y_2 X_3 NU - 2 Y_1 X_3 NU - X_1 Y_2 NU$$
$$+ 2 Y_1 X_2 NU + X_3 Y_3 - X_1 Y_3 - Y_2 X_3 + X_1 Y_2)(XMIN - XMAX)$$
$$(YMIN - YMAX)/2$$

ROW 4, COL 3

$$(X_2 Y_3 NU - X_1 Y_3 NU - 2 Y_2 X_3 NU + 2 Y_1 X_3 NU + X_2 Y_2 NU + X_1 Y_2 NU$$
$$- 2 Y_1 X_2 NU - X_2 Y_3 + X_1 Y_3 + X_2 Y_2 - X_1 Y_2)(XMIN - XMAX)$$
$$(YMIN - YMAX)/2$$

ROW 4, COL 4

$$- (Y_3^2 NU - 2 Y_2 Y_3 NU + Y_2^2 NU - Y_3^2 + 2 Y_2 Y_3 - 2 X_3^2 + 4 X_2 X_3 - Y_2^2$$
$$- 2 X_2^2)(XMIN - XMAX)(YMIN - YMAX)/2$$

ROW 5, COL 1

$$(X_3 Y_3 NU + X_2 Y_3 NU - 2 X_1 Y_3 NU - 2 Y_2 X_3 NU + Y_1 X_3 NU + 2 X_1 Y_2 NU$$
$$- Y_1 X_2 NU + X_3 Y_3 - X_2 Y_3 - Y_1 X_3 + Y_1 X_2)(XMIN - XMAX)$$
$$(YMIN - YMAX)/2$$

ROW 5, COL 2

$$(X_3 - X_1)(Y_3 - Y_1)(NU + 1)(XMIN - XMAX)(YMIN - YMAX)$$

$$- \frac{\qquad\qquad}{2}$$

ROW 5, COL 3

$$- (X_2 Y_3 NU - X_1 Y_3 NU - 2 Y_2 X_3 NU + 2 Y_1 X_3 NU + 2 X_1 Y_2 NU$$

$$- Y_1 X_2 NU - X_1 Y_1 NU - X_2 Y_3 + X_1 Y_3 + Y_1 X_2 - X_1 Y_1) (XMIN - XMAX)$$

$$(YMIN - YMAX)/2$$

ROW 5, COL 4

$$(Y_3^2 NU - Y_2 Y_3 NU - Y_1 Y_3 NU + Y_1 Y_2 NU - Y_3^2 + Y_2 Y_3 + Y_1 Y_3 - 2 X_3^2$$

$$+ 2 X_2 X_3 + 2 X_1 X_3 - Y_1 Y_2 - 2 X_1 X_2) (XMIN - XMAX) (YMIN - YMAX)/2$$

ROW 5, COL 5

$$- (Y_3^2 NU - 2 Y_1 Y_3 NU + Y_1^2 NU - Y_3^2 + 2 Y_1 Y_3 - 2 X_3^2 + 4 X_1 X_3 - Y_1^2$$

$$- 2 X_1^2) (XMIN - XMAX) (YMIN - YMAX)/2$$

ROW 6, COL 1

$$- (2 X_2 Y_3 NU - 2 X_1 Y_3 NU - Y_2 X_3 NU + Y_1 X_3 NU - X_2 Y_2 NU$$

$$+ 2 X_1 Y_2 NU - Y_1 X_2 NU + Y_2 X_3 - Y_1 X_3 - X_2 Y_2 + Y_1 X_2)$$

$$(XMIN - XMAX) (YMIN - YMAX)/2$$

ROW 6, COL 2

$$(2 X_2 Y_3 NU - 2 X_1 Y_3 NU - Y_2 X_3 NU + Y_1 X_3 NU + X_1 Y_2 NU - 2 Y_1 X_2 NU$$

$$+ X Y NU + Y X - Y X - X Y + X Y) (XMIN - XMAX)$$

ILLUSTRATIVE EXAMPLES

1 1   2 3   1 3   1 2   1 1

(YMIN - YMAX)/2

ROW 6, COL 3

$$\frac{(X_2 - X_1)\,(Y_2 - Y_1)\,(NU + 1)\,(XMIN - XMAX)\,(YMIN - YMAX)}{2}$$

ROW 6, COL 4

$$- (Y_2 Y_3 NU - Y_1 Y_3 NU - Y_2^2 NU + Y_1 Y_2 NU - Y_2 Y_3 + Y_1 Y_3 - 2 X_2 X_3 + 2 X_1 X_3 + Y_2^2 - Y_1 Y_2 + 2 X_2^2 - 2 X_1 X_2)\,(XMIN - XMAX)\,(YMIN - YMAX)/2$$

ROW 6, COL 5

$$(Y_2 Y_3 NU - Y_1 Y_3 NU - Y_1 Y_2 NU + Y_1^2 NU - Y_2 Y_3 + Y_1 Y_3 - 2 X_2 X_3 + 2 X_1 X_3 + Y_1 Y_2 + 2 X_1 X_2 - Y_1^2 - 2 X_1^2)\,(XMIN - XMAX)\,(YMIN - YMAX)/2$$

ROW 6, COL 6

$$- (Y_2^2 NU - 2 Y_1 Y_2 NU + Y_1^2 NU - Y_2^2 + 2 Y_1 Y_2 - 2 X_2^2 + 4 X_1 X_2 - Y_1^2 - 2 X_1^2)\,(XMIN - XMAX)\,(YMIN - YMAX)/2$$

< Display option 2 produces: >

```
The options for OUTPUTTING the STIFFNESS MATRIX are
     (1) Upper triangular portion in algebraic format
     (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED with YOUR SELECTION.
?
2;

Enter the FORTRAN ARRAY NAME for the STIFFNESS MATRIX
?
STIFF;

Multiply each coefficient by

        -E*T/(2*(X(2)*Y(3)-X(1)*Y(3)-Y(2)*X(3)+Y(1)*X(3)+X(1)*Y(2)-Y(1)*X(
     1  2))**2*(NU-1)*(NU+1))

The STIFFNESS MATRIX is

        STIFF(1,1) = -(X(3)**2*NU-2*X(2)*X(3)*NU+X(2)**2*NU-2*Y(3)**2+4*Y(
     1   2)*Y(3)-X(3)**2+2*X(2)*X(3)-2*Y(2)**2-X(2)**2)*(XMIN-XMAX)*(YMI
     2   N-YMAX)/2
        STIFF(2,1) = (X(3)**2*NU-X(2)*X(3)*NU-X(1)*X(3)*NU+X(1)*X(2)*NU-2*
     1   Y(3)**2+2*Y(2)*Y(3)+2*Y(1)*Y(3)-X(3)**2+X(2)*X(3)+X(1)*X(3)-2*Y
     2   (1)*Y(2)-X(1)*X(2))*(XMIN-XMAX)*(YMIN-YMAX)/2
        STIFF(3,1) = -(X(3)**2*NU-2*X(1)*X(3)*NU+X(1)**2*NU-2*Y(3)**2+4*Y(
     1   1)*Y(3)-X(3)**2+2*X(1)*X(3)-2*Y(1)**2-X(1)**2)*(XMIN-XMAX)*(YMI
     2   N-YMAX)/2
        STIFF(4,1) = -(X(2)*X(3)*NU-X(1)*X(3)*NU-X(2)**2*NU+X(1)*X(2)*NU-2
     1   *Y(2)*Y(3)+2*Y(1)*Y(3)-X(2)*X(3)+X(1)*X(3)+2*Y(2)**2-2*Y(1)*Y(2
     2   )+X(2)**2-X(1)*X(2))*(XMIN-XMAX)*(YMIN-YMAX)/2
        STIFF(5,1) = (X(2)*X(3)*NU-X(1)*X(3)*NU-X(1)*X(2)*NU+X(1)**2*NU-2*
     1   Y(2)*Y(3)+2*Y(1)*Y(3)-X(2)*X(3)+X(1)*X(3)+2*Y(1)*Y(2)+X(1)*X(2)
     2   -2*Y(1)**2-X(1)**2)*(XMIN-XMAX)*(YMIN-YMAX)/2
        STIFF(6,1) = -(X(2)**2*NU-2*X(1)*X(2)*NU+X(1)**2*NU-2*Y(2)**2+4*Y(
     1   1)*Y(2)-X(2)**2+2*X(1)*X(2)-2*Y(1)**2-X(1)**2)*(XMIN-XMAX)*(YMI
     2   N-YMAX)/2
        STIFF(7,1) = -(X(3)-X(2))*(Y(3)-Y(2))*(NU+1)*(XMIN-XMAX)*(YMIN-YMA
     1   X)/2
        STIFF(8,1) = (X(3)*Y(3)*NU-2*X(2)*Y(3)*NU+X(1)*Y(3)*NU+Y(2)*X(3)*N
     1   U-2*Y(1)*X(3)*NU-X(1)*Y(2)*NU+2*Y(1)*X(2)*NU+X(3)*Y(3)-X(1)*Y(3
     2   )-Y(2)*X(3)+X(1)*Y(2))*(XMIN-XMAX)*(YMIN-YMAX)/2
        STIFF(9,1) = (X(2)*Y(3)*NU-X(1)*Y(3)*NU-2*Y(2)*X(3)*NU+2*Y(1)*X(3)
     1   *NU+X(2)*Y(2)*NU+X(1)*Y(2)*NU-2*Y(1)*X(2)*NU-X(2)*Y(3)+X(1)*Y(3
     2   )+X(2)*Y(2)-X(1)*Y(2))*(XMIN-XMAX)*(YMIN-YMAX)/2
        STIFF(10,1) = -(Y(3)**2*NU-2*Y(2)*Y(3)*NU+Y(2)**2*NU-Y(3)**2+2*Y(2
     1   )*Y(3)-2*X(3)**2+4*X(2)*X(3)-Y(2)**2-2*X(2)**2)*(XMIN-XMAX)*(YM
     2   IN-YMAX)/2
        STIFF(11,1) = (X(3)*Y(3)*NU+X(2)*Y(3)*NU-2*X(1)*Y(3)*NU-2*Y(2)*X(3
     1   )*NU+Y(1)*X(3)*NU+2*X(1)*Y(2)*NU-Y(1)*X(2)*NU+X(3)*Y(3)-X(2)*Y(
     2   3)-Y(1)*X(3)+Y(1)*X(2))*(XMIN-XMAX)*(YMIN-YMAX)/2
        STIFF(12,1) = -(X(3)-X(1))*(Y(3)-Y(1))*(NU+1)*(XMIN-XMAX)*(YMIN-YM
```

```
1    AX)/2
 STIFF(13,1) = -(X(2)*Y(3)*NU-X(1)*Y(3)*NU-2*Y(2)*X(3)*NU+2*Y(1)*X(
1    3)*NU+2*X(1)*Y(2)*NU-Y(1)*X(2)*NU-X(1)*Y(1)*NU-X(2)*Y(3)+X(1)*Y
2    (3)+Y(1)*X(2)-X(1)*Y(1))*(XMIN-XMAX)*(YMIN-YMAX)/2
 STIFF(14,1) = (Y(3)**2*NU-Y(2)*Y(3)*NU-Y(1)*Y(3)*NU+Y(1)*Y(2)*NU-Y
1    (3)**2+Y(2)*Y(3)+Y(1)*Y(3)-2*X(3)**2+2*X(2)*X(3)+2*X(1)*X(3)-Y(
2    1)*Y(2)-2*X(1)*X(2))*(XMIN-XMAX)*(YMIN-YMAX)/2
 STIFF(15,1) = -(Y(3)**2*NU-2*Y(1)*Y(3)*NU+Y(1)**2*NU-Y(3)**2+2*Y(1
1    )*Y(3)-2*X(3)**2+4*X(1)*X(3)-Y(1)**2-2*X(1)**2)*(XMIN-XMAX)*(YM
2    IN-YMAX)/2
 STIFF(16,1) = -(2*X(2)*Y(3)*NU-2*X(1)*Y(3)*NU-Y(2)*X(3)*NU+Y(1)*X(
1    3)*NU-X(2)*Y(2)*NU+2*X(1)*Y(2)*NU-Y(1)*X(2)*NU+Y(2)*X(3)-Y(1)*X
2    (3)-X(2)*Y(2)+Y(1)*X(2))*(XMIN-XMAX)*(YMIN-YMAX)/2
 STIFF(17,1) = (2*X(2)*Y(3)*NU-2*X(1)*Y(3)*NU-Y(2)*X(3)*NU+Y(1)*X(3
1    )*NU+X(1)*Y(2)*NU-2*Y(1)*X(2)*NU+X(1)*Y(1)*NU+Y(2)*X(3)-Y(1)*X(
2    3)-X(1)*Y(2)+X(1)*Y(1))*(XMIN-XMAX)*(YMIN-YMAX)/2
 STIFF(18,1) = -(X(2)-X(1))*(Y(2)-Y(1))*(NU+1)*(XMIN-XMAX)*(YMIN-YM
1    AX)/2
 STIFF(19,1) = -(Y(2)*Y(3)*NU-Y(1)*Y(3)*NU-Y(2)**2*NU+Y(1)*Y(2)*NU-
1    Y(2)*Y(3)+Y(1)*Y(3)-2*X(2)*X(3)+2*X(1)*X(3)+Y(2)**2-Y(1)*Y(2)+2
2    *X(2)**2-2*X(1)*X(2))*(XMIN-XMAX)*(YMIN-YMAX)/2
 STIFF(20,1) = (Y(2)*Y(3)*NU-Y(1)*Y(3)*NU-Y(1)*Y(2)*NU+Y(1)**2*NU-Y
1    (2)*Y(3)+Y(1)*Y(3)-2*X(2)*X(3)+2*X(1)*X(3)+Y(1)*Y(2)+2*X(1)*X(2
2    )-Y(1)**2-2*X(1)**2)*(XMIN-XMAX)*(YMIN-YMAX)/2
 STIFF(21,1) = -(Y(2)**2*NU-2*Y(1)*Y(2)*NU+Y(1)**2*NU-Y(2)**2+2*Y(1
1    )*Y(2)-2*X(2)**2+4*X(1)*X(2)-Y(1)**2-2*X(1)**2)*(XMIN-XMAX)*(YM
2    IN-YMAX)/2
```

3.3.2 Isoparametric Formulation

```
* SYSTEM INITIALIZATION *

WELCOME TO ***** VERSION 1.0

     It is now   THURSDAY  DECEMBER 1,1977  2:18:4
     The current file is [SYINIT, FCN]
     The current device and username is [DSK, AK1G]

     Report problems to
     ALAN R. KORNCOFF
     DEPT. OF CIVIL ENGINEERING
     CARNEGIE-MELLON UNIVERSITY
     CMU-10A, AK1G

     Terminate all input with a SEMICOLON - ';'
     Input '?;' for HELP

     Input your LOGIN NAME
     ?
```

ILLUSTRATIVE EXAMPLES

```
AK1G;
Is AK1G correct ?  (YES; or NO;)
?
YES;

GREETINGS AK1G

 * METHOD SELECTION *

The available formulations include
    (1) THE ISOPARAMETRIC METHOD
    (2) THE GENERALIZED COORDINATE METHOD

Please enter the number of the method chosen (1 OR 2).
?
1;
ISOPARAMETRIC FORMULATION

 * ISOPARAMETRIC FORMULATION EXECUTIVE *

 * ISOPARARMETRIC FORMULATION INITIALIZATION *

Do you wish to set DUMP, BREAK or TRACE POINTS
?
NO;


 * PROBLEM PARAMETER SPECIFICATION - ISOPARAMETRIC *

Input the NUMBER OF ELEMENT NODES
?
3;

Input the NUMBER OF DEGREES OF FREEDOM PER NODE
?
2;

Input the NUMBER OF NATURAL COORDINATES
?
3;

Input the vector of the NAMES OF THE NATURAL COORDINATES
There should be 3 elements
ELEMENT 1 =
L1;
ELEMENT 2 =
L2;
ELEMENT 3 =
L3;

Input the vector of the NAMES OF THE GLOBAL COORDINATES
There should be 2 elements
ELEMENT 1 =
```

ILLUSTRATIVE EXAMPLES

```
X;
ELEMENT 2 =
Y;

Input the vector of the NAMES OF THE DISPLACEMENT VARIABLES
There should be 2 elements
ELEMENT 1 =
U;
ELEMENT 2 =
V;

 * SHAPE FUNCTION PROCESSOR - ISOPARAMETRIC *

ENTER the terms of the SHAPE FUNCTION ordered from
node 1 through node 3.
The 3 elements will be prompted for
ELEMENT 1 =
L1;
ELEMENT 2 =
L2;
ELEMENT 3 =
L3;

SHAPE FUNCTION MODIFICATION
The OPTIONS are
   (1) DISPLAY THE SHAPE FUNCTIONS
   (2) MODIFY THE SHAPE FUNCTIONS
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;

 * B MATRIX DATABASE GENERATION *

The OPTIONS for specifying STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) LIBRARY VALUES
Please ENTER the NUMBER ASSOCIATED WITH YOUR SELECTION
?
1;

< Though the following strain component specifications may be
automatically selected by chosing library option 3 or 4, the following
demonstrates the user input option. >

Input the NUMBER OF STRAIN COMPONENTS
?
3;
Input COMPONENT 1
?
D(U,X);
Input COMPONENT 2
```

ILLUSTRATIVE EXAMPLES

```
?
D(V,Y);
Input COMPONENT 3
?
D(U,Y) + D(V,X);


 * MATERIAL PROPERTIES SELECTION *

The options for the selection of the MATERIAL PROPERTIES MATRIX are
   (1) USER-SUPPLIED MATRIX
   (2) LIBRARY MATRIX
Please enter the NUMBER ASSOCIATED WITH YOUR SELECTION
?
1;

< Though the following material property specifications may be
automatically selected by chosing library option 3, the following
demonstrates the user input option. >

Please enter the UPPER TRIANGULAR PORTION of
the MATERIAL PROPERTIES MATRIX.
Each element will be prompted for with its ROW and COLUMN number.
You can also specify a CONSTANT SCALAR MULTIPLE.
ERRONEOUS INPUT may be corrected after the matrix is entered.

Do you wish to enter a CONSTANT SCALAR MULTIPLE (YES; OR NO;)
?
YES;
Enter the SCALAR MULTIPLE
?
E/(1-NU**2);

Enter the ELEMENTS of the MATERIAL PROPERTIES MATRIX
as they are prompted for
The matrix should be of order 3
ROW 1 , COLUMN 1 =
?
1;
ROW 1 , COLUMN 2 =
?
NU;
ROW 1 , COLUMN 3 =
?
0;
ROW 2 , COLUMN 2 =
?
1;
ROW 2 , COLUMN 3 =
?
0;
ROW 3 , COLUMN 3 =
?
```

ILLUSTRATIVE EXAMPLES

```
1-NU/2;

MATERIALS MATRIX MODIFICATION
The OPTIONS are
   (1) DISPLAY THE MATERIALS MATRIX
   (2) MODIFY THE MATERIALS MATRIX
   (3) DISPLAY THE MATERIALS MATRIX MULTIPLIER
   (4) MODIFY THE MATERIALS MATRIX MULTIPLIER
   (5) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPERATION
?
1;
THE MATRIX IS
                              [ 1   NU    0    ]
                              [                ]
                              [ NU  1     0    ]
                              [                ]
                              [            NU  ]
                              [ 0   0   1 --   ]
                              [            2   ]


MULTIPLIED BY

                                     E
                                  -------
                                        2
                                  1 - NU

Enter the NUMBER ASSOCIATED with the CHOSEN OPERATION
?
2;

MODIFY THE MATERIALSMATRIX
ONLY THE ELEMENTS ABOVE THE DIAGONAL NEED BE ENTERED

ENTER THE ROW NUMBER
?
3;
ENTER COLUMN NUMBER
?
3;
ELEMENT[3,3] WAS
                                      NU
                                  1 - --
                                      2

ENTER NEW VALUE
?
(1-NU)/2;
Enter the NUMBER ASSOCIATED with the CHOSEN OPERATION
?
5;
```

```
 * AUXILIARY TERM PROCESSOR *

ENTER ELEMENT VOLUME MODIFICATION FACTORS OR AUXILIARY TERMS
TYPE 'END;' to TERMINATE
?
T;
?
END;

AUXILIARY TERM MODIFICATION
The OPTIONS are
   (1) DISPLAY THE AUXILIARY TERMS
   (2) MODIFY AN AUXILIARY TERM
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;

 * INTEGRATION LIMITS - ISOPARAMETRIC *

Since the NATURAL COORDINATES specified are NOT INDEPENDENT,
you need ENTER only the LOWER LIMITS OF INTEGRATION

NATURAL COORDINATE, L1
UPPER LIMIT =
1
ENTER the LOWER LIMIT
?
0;
NATURAL COORDINATE, L2
UPPER LIMIT =
1 - L1
ENTER the LOWER LIMIT
?
1;
THIS LIMIT OF INTEGRATION MUST BE AN INTEGER, EITHER -1 OR 0.
PLEASE RE-ENTER.
?
0;

< A semantic check on  the input flagged an error in the specification
of the lower limit of integration. >

 * JACOBIAN GENERATION - ISOPARAMETRIC *

 * B MATRIX GENERATION - ISOPARAMETRIC *

 * INTEGRATION PROCESSING - ISOPARAMETRIC *

 * DISPLAY PRE-PROCESSOR - ISOPARAMETRIC *

 * DISPLAY PROCESSOR *
```

ILLUSTRATIVE EXAMPLES

The options for OUTPUTTING the STIFFNESS MATRIX are
(1) Upper triangular portion in algebraic format
(2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED with YOUR SELECTION.
?
1;

Multiply each coefficient by

$$\frac{E\,T}{-(X_2 Y_3 - X_1 Y_3 - Y_2 X_3 + Y_1 X_3 + X_1 Y_2 - Y_1 X_2)(NU - 1)(NU + 1)}$$

ROW 1, COL 1

$$-(X_3^2 NU - 2 X_2 X_3 NU + X_2^2 NU - 2 Y_3^2 + 4 Y_2 Y_3 - X_3^2 + 2 X_2 X_3 - 2 Y_2^2 - X_2^2)/4$$

ROW 2, COL 1

$$(X_3^2 NU - X_2 X_3 NU - X_1 X_3 NU + X_1 X_2 NU - 2 Y_3^2 + 2 Y_2 Y_3 + 2 Y_1 Y_3 - X_3^2 + X_2 X_3 + X_1 X_3 - 2 Y_1 Y_2 - X_1 X_2)/4$$

ROW 2, COL 2

$$-(X_3^2 NU - 2 X_1 X_3 NU + X_1^2 NU - 2 Y_3^2 + 4 Y_1 Y_3 - X_3^2 + 2 X_1 X_3 - 2 Y_1^2 - X_1^2)/4$$

ROW 3, COL 1

2

$$- (X_2 X_3 NU - X_1 X_3 NU - X_2 NU + X_1 X_2 NU - 2 Y_2 Y_3 + 2 Y_1 Y_3 - X_2 X_3$$

$$+ X_1 X_3 + 2 Y_2^2 - 2 Y_1 Y_2 + X_2^2 - X_1 X_2)/4$$

ROW 3, COL 2

$$(X_2 X_3 NU - X_1 X_3 NU - X_1 X_2 NU + X_1^2 NU - 2 Y_2 Y_3 + 2 Y_1 Y_3 - X_2 X_3$$

$$+ X_1 X_3 + 2 Y_1 Y_2 + X_1 X_2 - 2 Y_1^2 - X_1^2)/4$$

ROW 3, COL 3

$$- (X_2^2 NU - 2 X_1 X_2 NU + X_1^2 NU - 2 Y_2^2 + 4 Y_1 Y_2 - X_2^2 + 2 X_1 X_2 - 2 Y_1^2$$

$$- X_1^2)/4$$

ROW 4, COL 1

$$- \frac{(X_3 - X_2)(Y_3 - Y_2)(NU + 1)}{4}$$

ROW 4, COL 2

$$(X_3 Y_3 NU - 2 X_2 Y_3 NU + X_1 Y_3 NU + Y_2 X_3 NU - 2 Y_1 X_3 NU - X_1 Y_2 NU$$

$$+ 2 Y_1 X_2 NU + X_3 Y_3 - X_1 Y_3 - Y_2 X_3 + X_1 Y_2)/4$$

ROW 4, COL 3

$$(X_2 Y_3 NU - X_1 Y_3 NU - 2 Y_2 X_3 NU + 2 Y_1 X_3 NU + X_2 Y_2 NU + X_1 Y_2 NU$$

$$- 2 Y X NU - X Y + X Y + X Y - X Y)/4$$

ILLUSTRATIVE EXAMPLES

1 2 2 3 1 3 2 2 1 2

ROW 4, COL 4

$$- (Y_3^2 NU - 2 Y_2 Y_3 NU + Y_2^2 NU - Y_3^2 + 2 Y_2 Y_3 - 2 X_3^2 + 4 X_2 X_3 - Y_2^2 - 2 X_2^2)/4$$

ROW 5, COL 1

$$(X_3 Y_3 NU + X_2 Y_3 NU - 2 X_1 Y_3 NU - 2 Y_2 X_3 NU + Y_1 X_3 NU + 2 X_1 Y_2 NU - Y_1 X_2 NU + X_3 Y_3 - X_2 Y_3 - Y_1 X_3 + Y_1 X_2)/4$$

ROW 5, COL 2

$$- \frac{(X_3 - X_1) (Y_3 - Y_1) (NU + 1)}{4}$$

ROW 5, COL 3

$$- (X_2 Y_3 NU - X_1 Y_3 NU - 2 Y_2 X_3 NU + 2 Y_1 X_3 NU + 2 X_1 Y_2 NU - Y_1 X_2 NU - X_1 Y_1 NU - X_2 Y_3 + X_1 Y_3 + Y_1 X_2 - X_1 Y_1)/4$$

ROW 5, COL 4

$$(Y_3^2 NU - Y_2 Y_3 NU - Y_1 Y_3 NU + Y_1 Y_2 NU - Y_3^2 + Y_2 Y_3 + Y_1 Y_3 - 2 X_3^2 + 2 X_2 X_3 + 2 X_1 X_3 - Y_1 Y_2 - 2 X_1 X_2)/4$$

ROW 5, COL 5

2 2 2 2 2

ILLUSTRATIVE EXAMPLES

$$- (Y_3 NU - 2 Y_1 Y_3 NU + Y_1 NU - Y_3 + 2 Y_1 Y_3 - 2 X_3 + 4 X_1 X_3 - Y_1 - 2 X_1^2)/4$$

ROW 6, COL 1

$$- (2 X_2 Y_3 NU - 2 X_1 Y_3 NU - Y_2 X_3 NU + Y_1 X_3 NU - X_2 Y_2 NU$$

$$+ 2 X_1 Y_2 NU - Y_1 X_2 NU + Y_2 X_3 - Y_1 X_3 - X_2 Y_2 + Y_1 X_2)/4$$

ROW 6, COL 2

$$(2 X_2 Y_3 NU - 2 X_1 Y_3 NU - Y_2 X_3 NU + Y_1 X_3 NU + X_1 Y_2 NU - 2 Y_1 X_2 NU$$

$$+ X_1 Y_1 NU + Y_2 X_3 - Y_1 X_3 - X_1 Y_2 + X_1 Y_1)/4$$

ROW 6, COL 3

$$- \frac{(X_2 - X_1)(Y_2 - Y_1)(NU + 1)}{4}$$

ROW 6, COL 4

$$- (Y_2 Y_3 NU - Y_1 Y_3 NU - Y_2^2 NU + Y_1 Y_2 NU - Y_2 Y_3 + Y_1 Y_3 - 2 X_2 X_3$$

$$+ 2 X_1 X_3 + Y_2^2 - Y_1 Y_2 + 2 X_2^2 - 2 X_1 X_2)/4$$

ROW 6, COL 5

$$(Y_2 Y_3 NU - Y_1 Y_3 NU - Y_1 Y_2 NU + Y_1^2 NU - Y_2 Y_3 + Y_1 Y_3 - 2 X_2 X_3$$

$$+ 2 X X + Y Y + 2 X X - Y^2 - 2 X^2)/4$$

```
                                  1 3    1 2     1 2   1        1


ROW 6, COL 6

   2                      2       2            2            2
- (Y  NU - 2 Y  Y  NU + Y  NU - Y  + 2 Y  Y  - 2 X  + 4 X  X  - Y
   2          1  2        1       2      1  2      2       1  2   1

                                                                2
                                                           - 2 X )/4
                                                                1
```

< Display option 2 produces: >

```
The options for OUTPUTTING the STIFFNESS MATRIX are
    (1) Upper triangular portion in algebraic format
    (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED  with YOUR SELECTION.
?
2;

Enter the FORTRAN ARRAY NAME for the STIFFNESS MATRIX
?
STIFF;


Multiply each coefficient by

     -E*T/((X(2)*Y(3)-X(1)*Y(3)-Y(2)*X(3)+Y(1)*X(3)+X(1)*Y(2)-Y(1)*X(2)
    1    )*(NU-1)*(NU+1))

The STIFFNESS MATRIX is

      STIFF(1,1) = -(X(3)**2*NU-2*X(2)*X(3)*NU+X(2)**2*NU-2*Y(3)**2+4*Y(
    1    2)*Y(3)-X(3)**2+2*X(2)*X(3)-2*Y(2)**2-X(2)**2)/4
      STIFF(2,1) =  (X(3)**2*NU-X(2)*X(3)*NU-X(1)*X(3)*NU+X(1)*X(2)*NU-2*
    1    Y(3)**2+2*Y(2)*Y(3)+2*Y(1)*Y(3)-X(3)**2+X(2)*X(3)+X(1)*X(3)-2*Y
    2    (1)*Y(2)-X(1)*X(2))/4
      STIFF(3,1) = -(X(3)**2*NU-2*X(1)*X(3)*NU+X(1)**2*NU-2*Y(3)**2+4*Y(
    1    1)*Y(3)-X(3)**2+2*X(1)*X(3)-2*Y(1)**2-X(1)**2)/4
      STIFF(4,1) = -(X(2)*X(3)*NU-X(1)*X(3)*NU-X(2)**2*NU+X(1)*X(2)*NU-2
    1    *Y(2)*Y(3)+2*Y(1)*Y(3)-X(2)*X(3)+X(1)*X(3)+2*Y(2)**2-2*Y(1)*Y(2
    2    )+X(2)**2-X(1)*X(2))/4
      STIFF(5,1) =  (X(2)*X(3)*NU-X(1)*X(3)*NU-X(1)*X(2)*NU+X(1)**2*NU-2*
    1    Y(2)*Y(3)+2*Y(1)*Y(3)-X(2)*X(3)+X(1)*X(3)+2*Y(1)*Y(2)+X(1)*X(2)
    2    -2*Y(1)**2-X(1)**2)/4
      STIFF(6,1) = -(X(2)**2*NU-2*X(1)*X(2)*NU+X(1)**2*NU-2*Y(2)**2+4*Y(
    1    1)*Y(2)-X(2)**2+2*X(1)*X(2)-2*Y(1)**2-X(1)**2)/4
      STIFF(7,1) = -(X(3)-X(2))*(Y(3)-Y(2))*(NU+1)/4
      STIFF(8,1) =  (X(3)*Y(3)*NU-2*X(2)*Y(3)*NU+X(1)*Y(3)*NU+Y(2)*X(3)*N
```

```
     1      U-2*Y(1)*X(3)*NU-X(1)*Y(2)*NU+2*Y(1)*X(2)*NU+X(3)*Y(3)-X(1)*Y(3
     2      )-Y(2)*X(3)+X(1)*Y(2))/4
      STIFF(9,1) = (X(2)*Y(3)*NU-X(1)*Y(3)*NU-2*Y(2)*X(3)*NU+2*Y(1)*X(3)
     1      *NU+X(2)*Y(2)*NU+X(1)*Y(2)*NU-2*Y(1)*X(2)*NU-X(2)*Y(3)+X(1)*Y(3
     2      )+X(2)*Y(2)-X(1)*Y(2))/4
      STIFF(10,1) = -(Y(3)**2*NU-2*Y(2)*Y(3)*NU+Y(2)**2*NU-Y(3)**2+2*Y(2
     1      )*Y(3)-2*X(3)**2+4*X(2)*X(3)-Y(2)**2-2*X(2)**2)/4
      STIFF(11,1) = (X(3)*Y(3)*NU+X(2)*Y(3)*NU-2*X(1)*Y(3)*NU-2*Y(2)*X(3
     1      )*NU+Y(1)*X(3)*NU+2*X(1)*Y(2)*NU-Y(1)*X(2)*NU+X(3)*Y(3)-X(2)*Y(
     2      3)-Y(1)*X(3)+Y(1)*X(2))/4
      STIFF(12,1) = -(X(3)-X(1))*(Y(3)-Y(1))*(NU+1)/4
      STIFF(13,1) = -(X(2)*Y(3)*NU-X(1)*Y(3)*NU-2*Y(2)*X(3)*NU+2*Y(1)*X(
     1      3)*NU+2*X(1)*Y(2)*NU-Y(1)*X(2)*NU-X(1)*Y(1)*NU-X(2)*Y(3)+X(1)*Y
     2      (3)+Y(1)*X(2)-X(1)*Y(1))/4
      STIFF(14,1) = (Y(3)**2*NU-Y(2)*Y(3)*NU-Y(1)*Y(3)*NU+Y(1)*Y(2)*NU-Y
     1      (3)**2+Y(2)*Y(3)+Y(1)*Y(3)-2*X(3)**2+2*X(2)*X(3)+2*X(1)*X(3)-Y(
     2      1)*Y(2)-2*X(1)*X(2))/4
      STIFF(15,1) = -(Y(3)**2*NU-2*Y(1)*Y(3)*NU+Y(1)**2*NU-Y(3)**2+2*Y(1
     1      )*Y(3)-2*X(3)**2+4*X(1)*X(3)-Y(1)**2-2*X(1)**2)/4
      STIFF(16,1) = -(2*X(2)*Y(3)*NU-2*X(1)*Y(3)*NU-Y(2)*X(3)*NU+Y(1)*X(
     1      3)*NU-X(2)*Y(2)*NU+2*X(1)*Y(2)*NU-Y(1)*X(2)*NU+Y(2)*X(3)-Y(1)*X
     2      (3)-X(2)*Y(2)+Y(1)*X(2))/4
      STIFF(17,1) = (2*X(2)*Y(3)*NU-2*X(1)*Y(3)*NU-Y(2)*X(3)*NU+Y(1)*X(3
     1      )*NU+X(1)*Y(2)*NU-2*Y(1)*X(2)*NU+X(1)*Y(1)*NU+Y(2)*X(3)-Y(1)*X(
     2      3)-X(1)*Y(2)+X(1)*Y(1))/4
      STIFF(18,1) = -(X(2)-X(1))*(Y(2)-Y(1))*(NU+1)/4
      STIFF(19,1) = -(Y(2)*Y(3)*NU-Y(1)*Y(3)*NU-Y(2)**2*NU+Y(1)*Y(2)*NU-
     1      Y(2)*Y(3)+Y(1)*Y(3)-2*X(2)*X(3)+2*X(1)*X(3)+Y(2)**2-Y(1)*Y(2)+2
     2      *X(2)**2-2*X(1)*X(2))/4
      STIFF(20,1) = (Y(2)*Y(3)*NU-Y(1)*Y(3)*NU-Y(1)*Y(2)*NU+Y(1)**2*NU-Y
     1      (2)*Y(3)+Y(1)*Y(3)-2*X(2)*X(3)+2*X(1)*X(3)+Y(1)*Y(2)+2*X(1)*X(2
     2      )-Y(1)**2-2*X(1)**2)/4
      STIFF(21,1) = -(Y(2)**2*NU-2*Y(1)*Y(2)*NU+Y(1)**2*NU-Y(2)**2+2*Y(1
     1      )*Y(2)-2*X(2)**2+4*X(1)*X(2)-Y(1)**2-2*X(1)**2)/4
```

3.3.3 Comparison and Verification

< The verification for this case consists of comparing the results of both formulations with a fiducial vector, "RUBEV2", adapted from Rubenstein[14], pages 107-110. The element represented by RUBEV2 is located in space such that: X(1)=0, Y(1)=0, Y(2)=0, XMIN=0, YMIN=0, XMAX=X(2), YMAX=Y(3). For each formulation, the calculated vector is evaluated for these boundary conditions and subtracted from the fiducial vector thus producing a null vector. >

< For the generalized coordinate formulation: >

```
(D1)                              [DSK, AK1G]

(C2) LOADFILE(%3G,VALS);
```

ILLUSTRATIVE EXAMPLES

%3G VALS DSK AK1G being loaded
loading done
time= 2098 msec.
(D2) DONE

(C3) FUNCTIONAL[1,1];
time= 1 msec.

(D3)
$$- (X_3^2\,NU - 2\,X_2\,X_3\,NU + X_2^2\,NU - 2\,Y_3^2 + 4\,Y_2\,Y_3 - X_3^2 + 2\,X_2\,X_3 - 2\,Y_2^2 - X_2^2)\,(XMIN - XMAX)\,(YMIN - YMAX)/2$$

(C4) FACTOR;
time= 0 msec.

(D4)
$$- E\,T/(2\,(X_2\,Y_3 - X_1\,Y_3 - Y_2\,X_3 + Y_1\,X_3 + X_1\,Y_2 - Y_1\,X_2)^2\,(NU - 1)\,(NU + 1))$$

(C5) FTL:FUNCTIONAL%FACTOR$
time= 255 msec.

(C6) FTL[1,1];
time= 1 msec.

(D6)
$$E\,(X_3^2\,NU - 2\,X_2\,X_3\,NU + X_2^2\,NU - 2\,Y_3^2 + 4\,Y_2\,Y_3 - X_3^2 + 2\,X_2\,X_3 - 2\,Y_2^2 - X_2^2)\,T\,(XMIN - XMAX)\,(YMIN - YMAX)$$
$$/(4\,(X_2\,Y_3 - X_1\,Y_3 - Y_2\,X_3 + Y_1\,X_3 + X_1\,Y_2 - Y_1\,X_2)^2\,(NU - 1)\,(NU + 1))$$

< Expression (D6) is element (1,1) of the stiffness matrix generated for the generalized coordinate formulation. This matrix is stored under the name, "FTL". >

(C7) LOADFILE(CST,FID);

CST FID DSK AK1G being loaded
loading done
time= 1578 msec.
(D7) DONE

ILLUSTRATIVE EXAMPLES

(C8) RUBEV2[1,1];
time= 1 msec.

$$\text{(D8)} \quad \frac{E\left(\frac{(X_3 - X_2)^2 (1 - NU^2)}{2} + Y_3^2\, T\right)}{2\, X_2\, Y_3\, (1 - NU^2)}$$

< Expression (D8) is element (1,1) of the stiffness matrix in Rubenstein's example. This matrix is stored as "RUBEV2". >

< The generalized coordinate stiffness matrix is evaluated at the boundary conditions: >

(C9) FTL:EV(FTL,XMIN=0,XMAX=X[2],YMIN=0,YMAX=Y[3],X[1]=0,Y[1]=0,Y[2]=0)$
time= 3771 msec.

(C10) FTL[1,1];
time= 1 msec.

$$\text{(D10)} \quad \frac{E\,(X_3^2\, NU - 2\, X_2\, X_3\, NU + X_2^2\, NU - 2\, Y_3^2 - X_3^2 + 2\, X_2\, X_3 - X_2^2)\, T}{4\, X_2\, Y_3\, (NU - 1)\, (NU + 1)}$$

(C11) FTL[1,1]-RUBEV2[1,1]$
time= 9 msec.

(C12) RATSIMP(%);
time= 110 msec.
(D12) 0

(C13) FTL-RUBEV2$
time= 677 msec.

(C14) RATSIMP(%);
time= 2879 msec.

$$\text{(D14)} \quad MATRIX([0], [0], [0], [0], [0], [0], [0], [0], [0], [0],$$

$$[- ((X_3 + X_2)\, Y_3\, E\, NU + ((X_3 - X_2)\, Y_3 - 2\, Y_3 \left(\frac{(X_3 + X_2)\, NU + X_3 - X_2}{2}\right))$$

$$E)\, T/(4\, X_2\, Y_3\, NU^2 - 4\, X_2\, Y_3)], [0], [0], [0], [0], [0], [0], [0],$$

[0], [0], [0])

< Expression (D14) contains the term by term difference between the matrix generated by ***** and that in Rubenstein's example. For an undetermined reason, MACSYMA is unable to recognize that element 11 of this vector is equal to zero. To overcome this, the element is presented, in the following lines, in a format that makes this evaluation obvious by inspection. >

(C15) %[11,1];
time= 1 msec.

(D15) $$- ((X_3 + X_2) Y_3 E\ NU + ((X_3 - X_2) Y_3 - 2 Y_3 \left(\frac{(X_3 + X_2) NU + X_3 - X_2}{2}\right)) E) T/(4 X_2 Y_3 NU^2 - 4 X_2 Y_3)$$

(C16) NUM1(%);
time= 11 msec.

(D16) $$- ((X_3 + X_2) Y_3 E\ NU + ((X_3 - X_2) Y_3 - 2 Y_3 \left(\frac{(X_3 + X_2) NU + X_3 - X_2}{2}\right)) E) T$$

(C17) EV(%,E=1,T=1);
time= 28 msec.

(D17) $$- (X_3 + X_2) Y_3 NU + 2 Y_3 \left(\frac{(X_3 + X_2) NU + X_3 - X_2}{2}\right) - (X_3 - X_2) Y_3$$

(C18) EXPAND(%);
time= 25 msec.

(D18) $$- X_3 Y_3 NU - X_2 Y_3 NU + 2 Y_3 \left(\frac{X_3 NU}{2} + \frac{X_2 NU}{2} + \frac{X_3}{2} - \frac{X_2}{2}\right) - X_3 Y_3 + X_2 Y_3$$

< Expression (D18) is a factor of the numerator of the 11th element. After expanding the term in parenthesis, it can be seen that the following pairs of terms cancel: 1 and 3, 2 and 4, 5 and 7, and 6 and 8. Thus element 11 is indeed zero and the generated stiffness matrix is equivalent to the one presented by Rubenstein. >

ILLUSTRATIVE EXAMPLES

(C19) CLOSEFILE(%VER,C3G);

< For the isoparametric formulation an identical comparison is made between the generated matrix and Rubenstein's: >

(D1) [DSK, AK1G]

(C2) LOADFILE(%3I,VALS);

%3I VALS DSK AK1G being loaded
loading done
time= 1922 msec.
(D2) DONE

(C3) FTL:FUNCTIONAL%FACTOR$
time= 174 msec.

(C4) FTL[1,1];
time= 1 msec.

(D4)
$$\frac{E\,(X_3^2\,NU - 2\,X_2\,X_3\,NU + X_2^2\,NU - 2\,Y_3^2 + 4\,Y_2\,Y_3 - X_3^2 + 2\,X_2\,X_3 - 2\,Y_2^2 - X_2^2)\,T}{4\,(X_2\,Y_3 - X_1\,Y_3 - Y_2\,X_3 + Y_1\,X_3 + X_1\,Y_2 - Y_1\,X_2)\,(NU - 1)\,(NU + 1)}$$

< Expression (D4) is element (1,1) of the stiffness matrix generated for the isoparametric formulation. >

(C5) LOADFILE(CST,FID);

CST FID DSK AK1G being loaded
loading done
time= 1976 msec.
(D5) DONE

(C6) RUBEV2[1,1];
time= 1 msec.

(D6)
$$\frac{E\left(\dfrac{(X_3 - X_2)^2\,(1 - NU)}{2} + Y_3^2\right)T}{2\,X_2\,Y_3\,(1 - NU^2)}$$

< The isoparametric stiffness matrix is evaluated for the boundary

conditions. >

```
(C7) FTL:EV(FTL,X[1]=0,Y[1]=0,Y[2]=0)$

You have run out of LIST space.
Do you want more?
Type ALL; NONE; a level-no. or the name of a space.
LIST;
time= 3751 msec.

(C8) FTL[1,1];
time= 1 msec.
             2                     2          2    2                2
        E (X   NU - 2 X  X  NU + X   NU - 2 Y   - X   + 2 X  X  - X ) T
             3          2  3       2          3    3      2  3    2
(D8) ---------------------------------------------------------------------
                          4 X  Y  (NU - 1) (NU + 1)
                             2  3

(C9) FTL[1,1]-RUBEV2[1,1];
time= 9 msec.
             2                     2          2    2                2
        E (X   NU - 2 X  X  NU + X   NU - 2 Y   - X   + 2 X  X  - X ) T
             3          2  3       2          3    3      2  3    2
(D9) ---------------------------------------------------------------------
                          4 X  Y  (NU - 1) (NU + 1)
                             2  3

                                                              2
                                                  (X  - X )  (1 - NU)
                                                    3    2               2
                                               E (------------------ + Y ) T
                                                          2              3
                                             - -------------------------------
                                                                      2
                                                        2 X  Y  (1 - NU )
                                                           2  3

(C10) RATSIMP(%);
time= 111 msec.
(D10)                             0

(C11) FTL-RUBEV2$
time= 663 msec.

(C12) RATSIMP(%);
time= 2822 msec.
(D12) MATRIX([0], [0], [0], [0], [0], [0], [0], [0], [0], [0],

                                                           (X  + X ) NU + X  - X
                                                             3    2        3    2
[- ((X  + X ) Y  E NU + ((X  - X ) Y  - 2 Y (-----------------------))
      3    2   3          3    2   3      3             2
```

ILLUSTRATIVE EXAMPLES

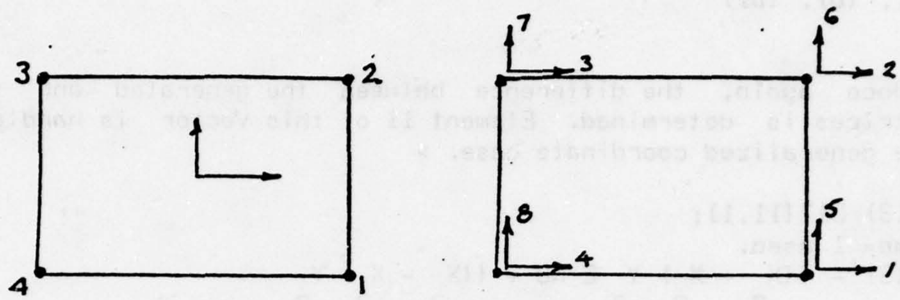
```
                               2
 E) T/(4 X  Y  NU  - 4 X  Y )], [0], [0], [0], [0], [0], [0], [0],
          2  3          2  3

[0], [0], [0])
```

< Once again, the difference between the generated and fiducial matrices is determined. Element 11 of this vector is handled as in the generalized coordinate case. >

```
(C13) D12[11,1];
time= 1 msec.
(D13) - ((X  + X ) Y  E NU + ((X  - X ) Y
           3    2   3            3    2   3

                (X  + X ) NU + X  - X
                  3    2        3    2                  2
         - 2 Y (----------------------)) E) T/(4 X  Y  NU  - 4 X  Y )
              3            2                      2  3            2  3

(C14) NUM(%);
time= 11 msec.
(D14) - ((X  + X ) Y  E NU + ((X  - X ) Y
           3    2   3            3    2   3
                                                 (X  + X ) NU + X  - X
                                                   3    2        3    2
                                          - 2 Y (----------------------)) E) T
                                               3            2

(C15) EV(%,E=1,T=1);
time= 29 msec.
                                   (X  + X ) NU + X  - X
                                     3    2        3    2
(D15) - (X  + X ) Y  NU + 2 Y (----------------------) - (X  - X ) Y
          3    2   3         3            2                 3    2   3

(C16) EXPAND(%);
time= 25 msec.
                                         X  NU   X  NU   X    X
                                          3       2       3    2
(D16) - X  Y  NU - X  Y  NU + 2 Y (----- + ----- + -- - --) - X  Y
         3  3        2  3          3   2       2      2    2     3  3

                                                                + X  Y
                                                                   2  3

(C17) CLOSEFILE(%VER,C31);
```

3.4 Four Node Quadrilateral

The nodal locations and the degrees of freedom for the four node quadrilateral are:



3.4.1 Generalized Coordinate Formulation

* SYSTEM INITIALIZATION *

WELCOME TO ***** VERSION 1.0

It is now WEDNESDAY DECEMBER 7,1977 11:28:58
The current file is [SYINIT, FCN]
The current device and username is [DSK, AK1G]

Report problems to
ALAN R. KORNCOFF
DEPT. OF CIVIL ENGINEERING
CARNEGIE-MELLON UNIVERSITY
CMU-10A, AK1G

Terminate all input with a SEMICOLON - ';'
Input '?;' for HELP

Input your LOGIN NAME
?
AK1G;
Is AK1G correct ? (YES; or NO;)
?
YES;

GREETINGS AK1G

* METHOD SELECTION *

The available formulations include
(1) THE ISOPARAMETRIC METHOD
(2) THE GENERALIZED COORDINATE METHOD

ILLUSTRATIVE EXAMPLES

```
Please enter the number of the method chosen (1 OR 2).
?
2;
GENERALIZED COORDINATE FORMULATION

  * GENERALIZED COORDINATE FORMULATION EXECUTIVE *

  * GENERALIZED COORDINATE FORMULATION INITIALIZATION *

Do you wish to set DUMP, BREAK or TRACE POINTS
?
NO;

  * PROBLEM PARAMETER SPECIFICATION - GENERALIZED COORDINATE *

Input the NUMBER OF ELEMENT NODES
?
4;

Input the NUMBER OF DEGREES OF FREEDOM PER NODE
?
2;

Input the vector of the NAMES OF THE GLOBAL COORDINATES
There should be 2 elements
ELEMENT 1 =
X;
ELEMENT 2 =
Y;

Input the vector of the NAMES OF THE DISPLACEMENT VARIABLES
There should be 2 elements
ELEMENT 1 =
U;
ELEMENT 2 =
V;

  * SHAPE FUNCTION PROCESSOR - GENERALIZED COORDINATE *

ENTER the terms of the SHAPE FUNCTION ordered from
GENERALIZED COORDINATE 1 through coordinate 4.
ELEMENT 1 =
1;
ELEMENT 2 =
X;
ELEMENT 3 =
Y;
ELEMENT 4 =
X*Y;

< This specification represents the displacement functions:
                    u = a1 + a2*x + a3*y + a4*x*y
```

```
                    v = a5 + a6*x + a7*y + a8*x*y
in which a1 through a8 are the generalized coordinates. >

SHAPE/FUNCTION MODIFICATION
The OPTIONS are
   (1) DISPLAY THE SHAPE FUNCTIONS
   (2) MODIFY THE SHAPE FUNCTIONS
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
1;

The terms of the SHAPE FUNCTION are
Term 1
                          1

Term 2
                          X

Term 3
                          Y

Term 4
                         X*Y

Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;


 * B MATRIX DATABASE GENERATION *

The OPTIONS for specifying STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) LIBRARY VALUES
Please ENTER the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) ONE DIMENSIONAL ELASTICITY
   (3) PLANE STRESS
   (4) PLANE STRAIN
   (5) AXISYMMETRIC
   (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
3;


 * MATERIAL PROPERTIES SELECTION *
```

```
ILLUSTRATIVE EXAMPLES

The options for the selection of the MATERIAL PROPERTIES MATRIX are
   (1) USER-SUPPLIED MATRIX
   (2) LIBRARY MATRIX
Please enter the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING MATERIAL PROPERTIES are
   (1) USER-SUPPLIED VALUES
   (2) ONE DIMENSIONAL ELASTICITY
   (3) PLANE STRESS
   (4) PLANE STRAIN
   (5) AXISYMMETRIC
   (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
3;


 * AUXILIARY TERM PROCESSOR *

ENTER ELEMENT VOLUME MODIFICATION FACTORS OR AUXILIARY TERMS
TYPE 'END;' to TERMINATE
?
T;
?
END;

AUXILIARY TERM MODIFICATION
The OPTIONS are
   (1) DISPLAY THE AUXILIARY TERMS
   (2) MODIFY AN AUXILIARY TERM
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;

 * INTEGRATION LIMITS - GENERALIZED COORDINATE *

 * INVERSION OF MATRIX, A *

 * B MATRIX GENERATION - GENERALIZED COORDINATE *

 * INTEGRATION PROCESSOR - GENERALIZED COORDINATE *

 * DISPLAY PRE-PROCESSOR - GENERALIZED COORDINATE *
```

< To avoid exceeding the list storage capacity of this experimental version of MACSYMA, the expression simplification function in the display pre-processor phase was bypassed. >

```
 * DISPLAY PROCESSOR *
```

< As the ouput is voluminous, only the first coefficent is displayed here. >

```
The options for OUTPUTTING the STIFFNESS MATRIX are
    (1) Upper triangular portion in algebraic format
    (2) FORTRAN CARD IMAGE format
ENTER the NUMBER ASSOCIATED  with YOUR SELECTION.
?
1;

Multiply each coefficient by
- E*T/(EXPT(X *Y *X *Y  - X *Y *X *Y  - Y *X *X *Y  + Y *X *X *Y
             2  3  4  4    1  3  4  4    2  3  4  4    1  3  4  4

  + X *Y *X *Y  - Y *X *X *Y  - X *X *Y *Y  + X *X *Y *Y  + X *Y *X *Y
     1  2  4  4    1  2  4  4    2  3  3  4    1  3  3  4    2  2  3  4

  - X *Y *X *Y  - X *X *Y *Y  + X *Y *X *Y  + Y *X *Y *X  - Y *X *Y *X
     1  1  3  4    1  2  2  4    1  1  2  4    2  3  3  4    1  3  3  4

  - X *Y *Y *X  + X *Y *Y *X  + Y *X *Y *X  - X *Y *Y *X  - X *Y *X *Y
     2  2  3  4    1  1  3  4    1  2  2  4    1  1  2  4    1  2  3  3

  + Y *X *X *Y  + X *X *Y *Y  - X *Y *X *Y  - Y *X *Y *X
     1  2  3  3    1  2  2  3    1  1  2  3    1  2  2  3

  + X *Y *Y *X , 2)*(NU - 1)*(NU + 1))
     1  1  2  3
```

< The MACSYMA function, EXPT, is used to represent a value equal to its first argument raised to the exponent given by its second argument.>

```
ROW 1, COL 1

                                                  2
(X *(Y  - Y ) - X *Y  - Y *(X  - X ) + Y *X )
  2   4    3     3  4    2   4    3     3  4

          3              3                  3                  3
  XMIN*YMIN     XMAX*YMIN     (1 - NU)*XMIN *YMIN   (1 - NU)*XMAX *YMIN
*(---------- - ---------- + ------------------- - -------------------
      3             3                6                    6

          3             3                  3
  XMIN*YMAX     XMAX*YMAX   (1 - NU)*XMIN *YMAX
- ---------- + ---------- - -------------------
      3             3                6

                  3
  (1 - NU)*XMAX *YMAX
+ -------------------) + 2*(X *(Y  - Y ) - X *Y  - Y *(X  - X )
          6                  2   4    3     3  4    2   4    3
```

$$+ Y_3 \cdot X_4) \cdot (Y_2 \cdot (X_4 \cdot Y_4 - X_3 \cdot Y_3) - X_2 \cdot Y_2 \cdot (Y_4 - Y_3) - Y_3 \cdot X_4 \cdot Y_4 + X_3 \cdot Y_3 \cdot Y_4)$$

$$\cdot (XMAX \cdot (\frac{YMAX^2}{2} - \frac{YMIN^2}{2}) - XMIN \cdot (\frac{YMAX^2}{2} - \frac{YMIN^2}{2}))$$

$$+ (- X_2 \cdot (X_4 \cdot Y_4 - X_3 \cdot Y_3) + X_3 \cdot X_4 \cdot Y_4 + X_2 \cdot Y_2 \cdot (X_4 - X_3) - X_3 \cdot Y_3 \cdot X_4)^2$$

$$\cdot (XMAX \cdot (\frac{(1 - NU) \cdot YMAX}{2} - \frac{(1 - NU) \cdot YMIN}{2})$$

$$- XMIN \cdot (\frac{(1 - NU) \cdot YMAX}{2} - \frac{(1 - NU) \cdot YMIN}{2}))$$

$$+ 2 \cdot (X_2 \cdot (Y_4 - Y_3) - X_3 \cdot Y_4 - Y_2 \cdot (X_4 - X_3) + Y_3 \cdot X_4)$$

$$\cdot (- X_2 \cdot (X_4 \cdot Y_4 - X_3 \cdot Y_3) + X_3 \cdot X_4 \cdot Y_4 + X_2 \cdot Y_2 \cdot (X_4 - X_3) - X_3 \cdot Y_3 \cdot X_4)$$

$$\cdot (\frac{(1 - NU) \cdot XMIN \cdot YMIN^2}{4} - \frac{(1 - NU) \cdot XMAX \cdot YMIN^2}{4} - \frac{(1 - NU) \cdot XMIN \cdot YMAX^2}{4}$$

$$+ \frac{(1 - NU) \cdot XMAX \cdot YMAX^2}{4})$$

$$+ (Y_2 \cdot (X_4 \cdot Y_4 - X_3 \cdot Y_3) - X_2 \cdot Y_2 \cdot (Y_4 - Y_3) - Y_3 \cdot X_4 \cdot Y_4 + X_3 \cdot Y_3 \cdot Y_4)^2$$

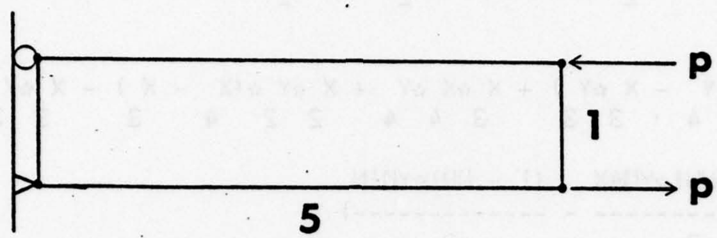
$$\cdot (XMAX \cdot (YMAX - YMIN) - XMIN \cdot (YMAX - YMIN))$$

3.4.2 Calibration of the Generalized Coordinate Results

Since there are no tabulated, closed-form expressions against which the generated coefficients could be compared, a calibration scheme was devised. The transverse deflection of the free end of a cantilever beam, as calculated using the generated stiffness matrix, is compared to the value determined using numerical integration as tabulated in Cook [15]. The cases of a moment and a transverse concentrated load applied to the free were considered:

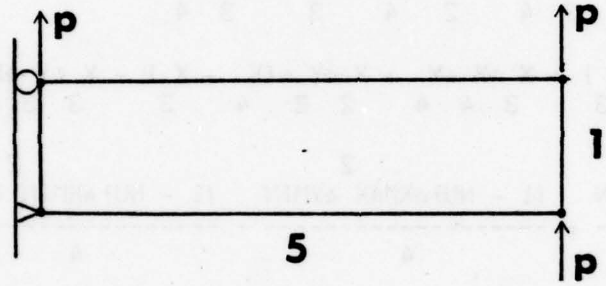
CASE 1: End Moment

The finite element model is shown below:



CASE 2: Transverse Load at Free end

The finite element model is shown below:



In both cases, the aspect ratio of the element is 5:1 and a unit thickness is assumed. Young's Modulus and Poisson's Ratio are taken as 1.0 and 0.25 respectively.

The stiffness matrix and common multiple (variables FUNCTIONAL and FACTOR) from the execution presented in section 3.4.1 were evaluated for the following geometric boundary conditions and material property values:

$$X(1) = X(2) = 5$$
$$Y(2) = Y(3) = 1$$
$$Y(1) = X(3) = X(4) = Y(4) = 0$$
$$XMAX = 5, \quad YMAX = 1, \quad XMIN = YMIN = 0$$
$$E = 1, \; NU = 0.25, \; T = 1$$

The displacement boundary conditions prescribe that degrees of freedom 3, 4 and 8 are restrained. The deflection in the direction of degree of freedom 5 is to be determined. The calculations are summarized below:

< FACTOR and FUNCTIONAL are evaluated at the geometric boundary conditions and for the specified material properties. >

```
(C7) FACTOR;
time= 0 msec.
(D7)                          1.70666666E-3

(C8) FUNCTIONAL;
time= 0 msec.
                              [  432.29167   ]
                              [              ]
                              [ - 369.79167  ]
                              [              ]
                              [  432.29167   ]
                              [              ]
                              [ - 216.145828 ]
                              [              ]
                              [  153.645828  ]
                              [              ]
                              [  432.29167   ]
                              [              ]
                              [  153.645828  ]
                              [              ]
                              [ - 216.145828 ]
                              [              ]
                              [ - 369.79167  ]
                              [              ]
                              [  432.29167   ]
                              [              ]
                              [  - 97.65625  ]
                              [              ]
                              [   19.53125   ]
                              [              ]
                              [   97.65625   ]
                              [              ]
                              [  - 19.53125  ]
                              [              ]
                              [  1057.29166  ]
                              [              ]
                              [  - 19.53125  ]
                              [              ]
                              [   97.65625   ]
                              [              ]
                              [   19.53125   ]
(D8)                          [              ]
                              [  - 97.65625  ]
                              [              ]
                              [ - 1033.85416 ]
                              [              ]
                              [  1057.29166  ]
                              [              ]
```

```
[    97.65625    ]
[                ]
[  - 19.53125    ]
[                ]
[  - 97.65625    ]
[                ]
[    19.53125    ]
[                ]
[  - 528.64584   ]
[                ]
[   505.208344   ]
[                ]
[  1057.29166    ]
[                ]
[    19.53125    ]
[                ]
[  - 97.65625    ]
[                ]
[  - 19.53125    ]
[                ]
[    97.65625    ]
[                ]
[   505.208344   ]
[                ]
[  - 528.64584   ]
[                ]
[  - 1033.85416  ]
[                ]
[  1057.29166    ]
```

< Each coefficient is multiplied by the common multiple. >

```
(C9) FUNCTIONAL:FACTOR(FUNCTIONAL)$
time= 71 msec.
```

< The vector is expanded into a square matrix. >

```
(C10) NUM:0$
time= 1 msec.

(C11) FOR I THRU 8 DO FOR J THRU I DO (NUM:NUM+1,FTL[I,J]:FTL[J,I]:FUN
CTIONAL[NUM,1]);
time= 701 msec.
(D11)                          DONE

(C12) STIFF:GENMATRIX(FTL,8,8);
time= 37 msec.
            [  0.737777784  ]         [ - 0.631111115 ]
            [               ]         [               ]
            [ - 0.631111115 ]         [  0.737777784  ]
            [               ]         [               ]
            [ - 0.36888888  ]         [  0.26222221   ]
            [               ]         [               ]
```

```
                [ 0.26222221   ]       [ - 0.36888888  ]
(D12)  Col 1 =  [              ] Col 2 = [             ]
                [ - 0.166666666 ]      [ 0.033333333   ]
                [              ]       [               ]
                [ - 0.033333333 ]      [ 0.166666666   ]
                [              ]       [               ]
                [ 0.166666666  ]       [ - 0.033333333 ]
                [              ]       [               ]
                [ 0.033333333  ]       [ - 0.166666666 ]

         [ - 0.36888888  ]       [ 0.26222221    ]
         [               ]       [               ]
         [ 0.26222221    ]       [ - 0.36888888  ]
         [               ]       [               ]
         [ 0.737777784   ]       [ - 0.631111115 ]
         [               ]       [               ]
         [ - 0.631111115 ]       [ 0.737777784   ]
 Col 3 = [               ] Col 4 = [             ]
         [ 0.166666666   ]       [ - 0.033333333 ]
         [               ]       [               ]
         [ 0.033333333   ]       [ - 0.166666666 ]
         [               ]       [               ]
         [ - 0.166666666 ]       [ 0.033333333   ]
         [               ]       [               ]
         [ - 0.033333333 ]       [ 0.166666666   ]

         [ - 0.166666666 ]       [ - 0.033333333 ]
         [               ]       [               ]
         [ 0.033333333   ]       [ 0.166666666   ]
         [               ]       [               ]
         [ 0.166666666   ]       [ 0.033333333   ]
         [               ]       [               ]
         [ - 0.033333333 ]       [ - 0.166666666 ]
 Col 5 = [               ] Col 6 = [             ]
         [ 1.80444442    ]       [ - 1.76444443  ]
         [               ]       [               ]
         [ - 1.76444443  ]       [ 1.80444442    ]
         [               ]       [               ]
         [ - 0.90222224  ]       [ 0.86222224    ]
         [               ]       [               ]
         [ 0.86222224    ]       [ - 0.90222224  ]

         [ 0.166666666   ]       [ 0.033333333   ]
         [               ]       [               ]
         [ - 0.033333333 ]       [ - 0.166666666 ]
         [               ]       [               ]
         [ - 0.166666666 ]       [ - 0.033333333 ]
         [               ]       [               ]
         [ 0.033333333   ]       [ 0.166666666   ]
 Col 7 = [               ] Col 8 = [             ]
         [ - 0.90222224  ]       [ 0.86222224    ]
         [               ]       [               ]
         [ 0.86222224    ]       [ - 0.90222224  ]
```

```
          [                ]          [                ]
          [  1.80444442    ]          [ - 1.76444443   ]
          [                ]          [                ]
          [ - 1.76444443   ]          [  1.80444442    ]
```

< The matrix is corrected for the displacement boundary conditions by eliminating rows and columns 3, 4 and 8. >

```
(C13) STIFFREDUCED:SUBMATRIX(3,4,8,STIFF,3,4,8);
time= 8 msec.
                  [  0.737777784   ]          [ - 0.631111115 ]
                  [                ]          [               ]
                  [ - 0.631111115  ]          [  0.737777784  ]
                  [                ]          [               ]
(D13)   Col 1 =   [ - 0.166666666  ]  Col 2 = [  0.033333333  ]
                  [                ]          [               ]
                  [ - 0.033333333  ]          [  0.166666666  ]
                  [                ]          [               ]
                  [  0.166666666   ]          [ - 0.033333333 ]

          [ - 0.166666666 ]          [ - 0.033333333 ]
          [               ]          [               ]
          [  0.033333333  ]          [  0.166666666  ]
          [               ]          [               ]
  Col 3 = [  1.80444442   ]  Col 4 = [ - 1.76444443  ]
          [               ]          [               ]
          [ - 1.76444443  ]          [  1.80444442   ]
          [               ]          [               ]
          [ - 0.90222224  ]          [  0.86222224   ]

          [  0.166666666  ]
          [               ]
          [ - 0.033333333 ]
          [               ]
  Col 5 = [ - 0.90222224  ]
          [               ]
          [  0.86222224   ]
          [               ]
          [  1.80444442   ]
```

< The reduced matrix is inverted. MACSYMA logs its use of function RAT. >

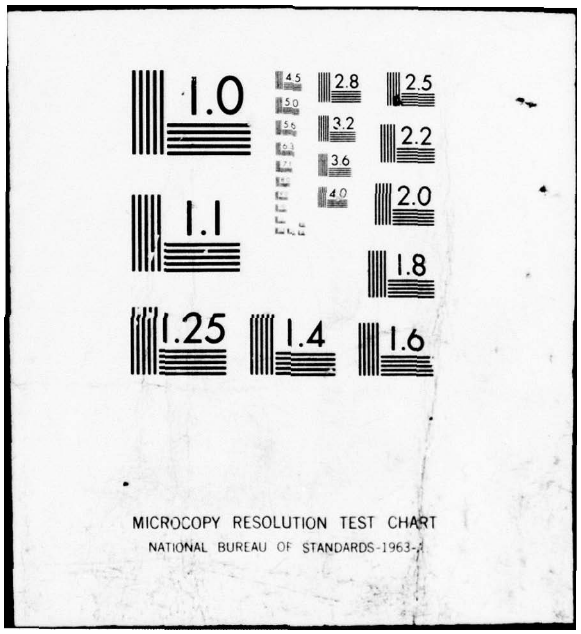
```
(C14) RESULT:EV(STIFFREDUCED↑↑-1,NUMER:TRUE);
RAT replaced 0.737777784 by 166/225 = 0.73777778
RAT replaced -0.631111115 by -142/225 = -0.63111111
RAT replaced -0.166666666 by -1/6 = -0.166666666
RAT replaced -0.033333333 by -1/30 = -0.033333333
RAT replaced 0.166666666 by 1/6 = 0.166666666
RAT replaced -0.631111115 by -142/225 = -0.63111111
RAT replaced 0.737777784 by 166/225 = 0.73777778
RAT replaced 0.033333333 by 1/30 = 0.033333333
RAT replaced 0.166666666 by 1/6 = 0.166666666
```

AD-A052 561 CARNEGIE INST OF TECH PITTSBURGH PA DEPT OF CIVIL EN--ETC F/G 12/1
SYMBOLIC GENERATION OF FINITE ELEMENT STIFFNESS MATRICES. (U)
JAN 78 A R KORNOFF, S J FENVES N00014-76-C-0345

UNCLASSIFIED R-78-103 NL

2 OF 2
AD
A052561

END
DATE
FILMED
5-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-

```
RAT replaced -0.033333333 by -1/30 = -0.033333333
RAT replaced -0.166666666 by -1/6 = -0.166666666
RAT replaced 0.033333333 by 1/30 = 0.033333333
RAT replaced 1.80444442 by 406/225 = 1.80444445
RAT replaced -1.76444443 by -397/225 = -1.76444444
RAT replaced -0.90222224 by -203/225 = -0.90222222
RAT replaced -0.033333333 by -1/30 = -0.033333333
RAT replaced 0.166666666 by 1/6 = 0.166666666
RAT replaced -1.76444443 by -397/225 = -1.76444444
RAT replaced 1.80444442 by 406/225 = 1.80444445
RAT replaced 0.86222224 by 194/225 = 0.862222224
RAT replaced 0.166666666 by 1/6 = 0.166666666
RAT replaced -0.033333333 by -1/30 = -0.033333333
RAT replaced -0.90222224 by -203/225 = -0.90222222
RAT replaced 0.86222224 by 194/225 = 0.862222224
RAT replaced 1.80444442 by 406/225 = 1.80444445
time= 469 msec.
(D14)

      [ 6.35542166   3.6445783    6.77710843   6.52710843   - 0.25      ]
      [                                                                 ]
      [ 3.6445783    6.35542166  - 6.77710843 - 7.02710843  - 0.25      ]
      [                                                                 ]
      [ 6.77710843  - 6.77710843  46.939729    46.3855424   0.55418719  ]
      [                                                                 ]
      [ 6.52710843  - 7.02710843  46.3855424   46.585542    0.2         ]
      [                                                                 ]
      [  - 0.25      - 0.25       0.55418719   0.2          0.75418719  ]
```

< Degree of freedom 5 is represented by the third row of the reduced matrix. >

< For case 1, unit loads are applied in the positive and negative direction of degrees of freedom 1 and 2 respectively. >

```
(C15) DELTA1:RESULT[3,1]-RESULT[3,2];
time= 3 msec.
(D15)                        13.5542169
```

< For case 2, unit loads are applied in the positive direction of degrees of freedom 5 and 6 and in the negative direction of degree of freedom 7. >

```
(C16) DELTA2:RESULT[3,3]+RESULT[3,4]-RESULT[3,5];
time= 4 msec.
(D16)                        92.771085
```

In reference [15] Cook scales the results from the finite element model with respect to a normalized value generated by applying beam theory. The calculated deflections of the free end of the cantilever subjected to both loading conditions are tabulated below. The values presented are: (a) Cook's scaled result from the finite element model; (b) Cook's normalized value from beam theory; (c) The value determined from the generated stiffness matrix and (d) The un-normalized value caluculated from beam theory.

CASE 1:
(a) Cook's finite element model: 9.0
(b) Cook's normalized beam theory value: 100.0
(c) Result from generated stiffness matrix: 13.55
(d) Un-normalized beam theory result: 150.0

Scaling the results form the generated stiffness matrix produces

$$13.55 * 100.0 / 150.0 = 9.03$$

which agrees with Cook's results for the finite element model.

CASE 2:
(a) Cook's finite element model: 9.3
(b) Cook's normalized beam theory value: 102.6
(c) Result from generated stiffness matrix: 92.8
(d) Un-normalized beam theory result:
(deflection due to bending =) 1000
+ (deflection due to shear =) 30
= 1030

Scaling the results from the generated stiffness matrix produces

$$92.8 * 102.6 / 1030.0 = 9.2$$

which agrees well with Cook's results for the finite element model. The deviation here is most likely due to the difference in the values calculated using beam theory.

Thus, the values calculated using the generated template agree quite well with the tabulated results.

3.4.3 Isoparametric Formulation

```
* SYSTEM INITIALIZATION *

WELCOME TO ***** VERSION 1.0

It is now   WEDNESDAY  DECEMBER 7,1977  10:0:7
The current file is {SYINIT, FCN}
The current device and username is [DSK, AK1G]

Report problems to
ALAN R. KORNCOFF
DEPT. OF CIVIL ENGINEERING
CARNEGIE-MELLON UNIVERSITY
CMU-10A, AK1G
```

ILLUSTRATIVE EXAMPLES

```
Terminate all input with a SEMICOLON - ';'
Input '?;' for HELP

Input your LOGIN NAME
?
AK1G;
Is AK1G correct ?  (YES; or NO;)
?
YES;

GREETINGS AK1G

 * METHOD SELECTION *

The available formulations include
   (1) THE ISOPARAMETRIC METHOD
   (2) THE GENERALIZED COORDINATE METHOD

Please enter the number of the method chosen (1 OR 2).
?
1;
ISOPARAMETRIC FORMULATION

 * ISOPARAMETRIC FORMULATION EXECUTIVE *

 * ISOPARARMETRIC FORMULATION INITIALIZATION *

Do you wish to set DUMP, BREAK or TRACE POINTS
?
YES;

1. ISOEIN      2. INPISO     3. SFNISO     4. BMDATA
5. MATERL      6. AUXTER     7. LIMTIS     8. JACOBN
9. BMXISO      10. INTISO     11. DISIPP     12. DISPLY
SET DUMP POINTS
Designate selected PHASES by entering
the associated integer INDEX or 'ALL' for all phases.
Type 'END' to TERMINATE.
?
END;

1. ISOEIN      2. INPISO     3. SFNISO     4. BMDATA
5. MATERL      6. AUXTER     7. LIMTIS     8. JACOBN
9. BMXISO      10. INTISO     11. DISIPP     12. DISPLY
SET BREAKPOINTS
Designate selected PHASES by entering
the associated integer INDEX or 'ALL' for all phases.
Type 'END' to TERMINATE.
?
9;
?
END;
```

```
1. ISOEIN      2. INPISO      3. SFNISO      4. BMDATA
5. MATERL      6. AUXTER      7. LIMTIS      8. JACOBN
9. BMXISO      10. INTISO     11. DISIPP     12. DISPLY
SET TRACE POINTS
Designate selected PHASES by entering
the associated integer INDEX or 'ALL' for all phases.
Type 'END' to TERMINATE.
?
END;

 * PROBLEM PARAMETER SPECIFICATION - ISOPARAMETRIC *

Input the NUMBER OF ELEMENT NODES
?
4;

Input the NUMBER OF DEGREES OF FREEDOM PER NODE
?
2;

Input the NUMBER OF NATURAL COORDINATES
?
2;

Input the vector of the NAMES OF THE NATURAL COORDINATES
There should be 2 elements
ELEMENT 1 =
S;
ELEMENT 2 =
T;

Input the vector of the NAMES OF THE GLOBAL COORDINATES
There should be 2 elements
ELEMENT 1 =
X;
ELEMENT 2 =
Y;

Input the vector of the NAMES OF THE DISPLACEMENT VARIABLES
There should be 2 elements
ELEMENT 1 =
U;
ELEMENT 2 =
V;

 * SHAPE FUNCTION PROCESSOR - ISOPARAMETRIC *

ENTER the terms of the SHAPE FUNCTION ordered from
node 1 through node 4.
The 4 elements will be prompted for
ELEMENT 1 =
```

```
1/4*(1+S)*(1-T);
ELEMENT 2 =
1/4*(1+S)*(1+T);
ELEMENT 3 =
1/4*(1-S)*(1+T);
ELEMENT 4 =
1/4*(1-S)*(1-T);
                    :
SHAPE FUNCTION MODIFICATION
The OPTIONS are
   (1) DISPLAY THE SHAPE FUNCTIONS
   (2) MODIFY THE SHAPE FUNCTIONS
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
1;

The terms of the SHAPE FUNCTION are
Term 1
                         (S + 1)*(1 - T)
                         ---------------
                                4

Term 2
                         (S + 1)*(T + 1)
                         ---------------
                                4

Term 3
                         (1 - S)*(T + 1)
                         ---------------
                                4

Term 4
                         (1 - S)*(1 - T)
                         ---------------
                                4

Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;


 * B MATRIX DATABASE GENERATION *

The OPTIONS for specifying STRAIN COMPONENTS are
   (1) USER-SUPPLIED VALUES
   (2) LIBRARY VALUES
Please ENTER the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING STRAIN COMPONENTS are
```

ILLUSTRATIVE EXAMPLES

```
   (1) USER-SUPPLIED VALUES
   (2) ONE DIMENSIONAL ELASTICITY
   (3) PLANE STRESS
   (4) PLANE STRAIN
   (5) AXISYMMETRIC
   (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
3;


 * MATERIAL PROPERTIES SELECTION *

The options for the selection of the MATERIAL PROPERTIES MATRIX are
   (1) USER-SUPPLIED MATRIX
   (2) LIBRARY MATRIX
Please enter the NUMBER ASSOCIATED WITH YOUR SELECTION
?
2;

The LIBRARY OPTIONS for SPECIFYING MATERIAL PROPERTIES are
   (1) USER-SUPPLIED VALUES
   (2) ONE DIMENSIONAL ELASTICITY
   (3) PLANE STRESS
   (4) PLANE STRAIN
   (5) AXISYMMETRIC
   (6) LINEAR ISOTROPIC ELASTICITY - 3D
Please ENTER the NUMBER ASSOCIATED WITH YOUR CHOICE
?
3;


 * AUXILIARY TERM PROCESSOR *

ENTER ELEMENT VOLUME MODIFICATION FACTORS OR AUXILIARY TERMS
TYPE 'END;' to TERMINATE
?
TH;
?
END;

AUXILIARY TERM MODIFICATION
The OPTIONS are
   (1) DISPLAY THE AUXILIARY TERMS
   (2) MODIFY AN AUXILIARY TERM
   (3) TERMINATE THIS FUNCTION
Enter the NUMBER ASSOCIATED with the CHOSEN OPTION
?
3;


 * INTEGRATION LIMITS - ISOPARAMETRIC *
```

ILLUSTRATIVE EXAMPLES

```
ENTER the LIMITS OF INTEGRATION for NATURAL COORDINATE, S
LOWER LIMIT =
?
-1;
UPPER LIMIT =
?
1;
ENTER the LIMITS OF INTEGRATION for NATURAL COORDINATE, T
LOWER LIMIT =
?
-1;
UPPER LIMIT =
?
1;

 * JACOBIAN GENERATION - ISOPARAMETRIC *

 * B MATRIX GENERATION - ISOPARAMETRIC *


BREAKPOINT FOR PHASE BMXISO ENCOUNTERED
TYPE 'EXIT;' TO RESUME

35 msec.
(MACSYMA-BREAK)
←CLOSEFILE(AK1G,IQCMD);
```

< To avoid exceeding storage space provided by the present version of MACSYMA, processing was suspended after the determination of the {B} matrix. The record file was closed, generated values were saved on file (AK1G,IQVALS), the current job was terminated and a new job was initiated. The values were then restored and selected functions from the integration processor and display pre-processor phases were then 9used to produce the integrand of equation (2) (section 1.2.1): >

```
(D2)                                [DSK, AK1G]

(C3) LOADFILE(AK1G,IQVALS);

AK1G IQVALS DSK AK1G being loaded
loading done
(D3)                                DONE

(C4) LOADFILE(INTISO,FCN);

INTISO FCN DSK AK1G being loaded
loading done
(D4)                                DONE
```

ILLUSTRATIVE EXAMPLES

```
(C5) LOADFILE(DISIPP,FCN);
DISIPP FCN DSK AK1G being loaded
loading done
(D5)                                   DONE

(C6) BMATRIX:FACTOR(BMATRIX)$

(C7) FUNCTIONAL:QUADRATICFORM(BMATRIX,MATERIALSMATRIX)$

GENMAT FASL DSK MAXOUT being loaded
loading done

(C8) FUNCTIONAL[1,1];
```

$$(D8)\quad \frac{(Y_4 T - Y_3 T - Y_3 S + Y_2 S - Y_4 + Y_2)^2}{64} + \frac{(1 - NU)\,(X_4 T - X_3 T - X_3 S + X_2 S - X_4 + X_2)^2}{128}$$

< Expression (D8) represents the coefficient before it is multiplied by any auxiliary terms which are functions of the natural coordinates. >

```
(C9) INTVARSINSTACK(NATURALCOORDS,STACK,STACKPOINTER);
(D9)                          DONE

(C10) FUNCTIONAL:FUNCTIONALMULTINTVARS(STACK,STACKPOINTER,FUNCTIONAL)$

(C11) FUNCTIONAL[1,1];
```

$$(D11)\quad - 8 \left(\frac{(Y_4 T - Y_3 T - Y_3 S + Y_2 S - Y_4 + Y_2)^2}{64} + \frac{(1 - NU)\,(X_4 T - X_3 T - X_3 S + X_2 S - X_4 + X_2)^2}{128}\right)$$

$$/(((X_2 - X_1) Y_4 + (Y_1 - Y_2) X_4 + (X_1 - X_2) Y_3 + (Y_2 - Y_1) X_3) T$$

$$+ ((X_3 - X_2) Y_4 + (Y_2 - Y_3) X_4 + X_1 Y_3 - Y_1 X_3 - X_1 Y_2 + Y_1 X_2) S$$

```
+ (X  - X ) Y  + (Y  - Y ) X  - X Y  + Y X  - X Y  + Y X )
    1    3   4     3    1   4    2 3    2 3    1 2    1 2

(C12) FACTOR:MULTFACTORISO(STACK,STACKPOINTER);
                               E TH
(D12)                 - -----------------
                       (NU - 1) (NU + 1)

(C13) CLOSEFILE(IQCMD,DISP);
```

< As the output is voluminous, only the first coefficient (expression (D11)) and the common multiple (expression (D12)) are displayed. >

< It is to be noted that although integration could not be achieved, because of the restrictions placed upon storage space in this experimental version of MACSYMA, the results produced are in the format appropriate for the hybrid symbolic-numeric method dicussed in section 2.1 . That is, a "matrix template" could be produced for expressions such as (D11); any numerical quadrature program could call such a template, supplying numerical values for the element coordinates as well as for the natural coordinates s and t at the selected quadrature points. >

3.5 Algebraic Format of Generated Expressions

MACSYMA performs no automatic simplification of symbolic expressions. Instead, the user is provided with a class of functions, with which he can interactively produce output formats to his liking. Computation time is generally high for these operations.

It would be a major research effort, in itself, to invest ***** with a set of heuristics that would enable it to make decisions concerning "optimal" output formats. Current results are produced by applying the FACTOR function to the output of the integration processors.

Some of the various output formats which may be generated are demonstrated in the run below. Several equivalent forms are presented in algebraic and FORTRAN card image format. The example employs element [1,1] of the factored stiffness matrix for the CST, generalized coordinate formulation.

(D2) [DSK, AK1G]

(C3) LOADFILE(%3I,VALS);

%3I VALS DSK AK1G being loaded
loading done
(D3) DONE

< Expresions (D4) and (D5) are those currently produced by *****. >

(C4) FTL11:FUNCTIONAL[1,1];

$$(D4)\quad -(X_3^2\ NU - 2\ X_2\ X_3\ NU + X_2^2\ NU - 2\ Y_3^2 + 4\ Y_2\ Y_3 - X_3^2 + 2\ X_2\ X_3 - 2\ Y_2^2 - X_2^2)/4$$

(C5) FORTRAN(%);

```
FORTRA FASL DSK MACSYM being loaded
loading done
      -(X(3)**2*NU-2*X(2)*X(3)*NU+X(2)**2*NU-2*Y(3)**2+4*Y(2)*Y(3)-X(3)*
     1 *2+2*X(2)*X(3)-2*Y(2)**2-X(2)**2)/4
(D5)                                DONE
```

< The application of the EXPAND function will cause products of sums and exponentiated sums to be multiplied out, numerators of rational expressions which are sums to be split into their respective terms, and multiplication to be distributed over addition at all levels of its argument. [10] >

```
(C6) EXPAND(FTL11);
              2              2    2              2        2   2
            X  NU   X  X  NU   X  NU   Y          X   X  X    Y   X
             3       2  3       2       3          3   2  3    2   2
(D6) - ----- + -------- - ----- + -- - Y  Y  + -- - ----- + -- + --
           4        2        4     2    2  3  4       2     2    4

(C7) FORTRAN(%);
      -X(3)**2*NU/4+X(2)*X(3)*NU/2-X(2)**2*NU/4+Y(3)**2/2-Y(2)*Y(3)+X(3)
     1    **2/4-X(2)*X(3)/2+Y(2)**2/2+X(2)**2/4
(D7)                                DONE
```

< RATSIMP "rationally" simplifies its argument and all of its subexpressions including arguments to non-rational functions. The result is returned as the quotient of two polynomials in a recursive form, i.e. the coefficients of the main variable are polynomials in the other variables. [11] >

```
(C8) RATSIMP(FTL11);
(D8)

      2            2        2             2                2    2
   (X   - 2 X  X   + X ) NU - 2 Y   + 4 Y  Y  - X   + 2 X  X  - 2 Y  - X
     3       2  3    2           3       2  3    3       2  3      2    2
 - ---------------------------------------------------------------------
                                   4

(C9) FORTRAN(%);
      -((X(3)**2-2*X(2)*X(3)+X(2)**2)*NU-2*Y(3)**2+4*Y(2)*Y(3)-X(3)**2+2
     1    *X(2)*X(3)-2*Y(2)**2-X(2)**2)/4
(D9)                                DONE
```

< Horner(exp) will convert exp into a rearranged representation as in Horner's rule. [12] >

```
(C10) HORNER(FTL11);

OPTIM FASL DSK MAXOUT being loaded
loading done
                                2
(D10) (((2 X  - X ) X  - X ) NU + Y  (2 Y  - 4 Y ) + X  (X  - 2 X )
            2    3   3    2          3     3      2    3   3     2

                                                          2    2
                                                    + 2 Y  + X )/4
                                                         2    2

(C11) FORTRAN(%);
      (((2*X(2)-X(3))*X(3)-X(2)**2)*NU+Y(3)*(2*Y(3)-4*Y(2))+X(3)*(X(3)-2
     1    *X(2))+2*Y(2)**2+X(2)**2)/4
(D11)                               DONE
```

< It is to be noted that the range of the number of multiplications

and divisions extends from 13 using the HORNER function up to 20 produced by EXPAND. >

```
(C12) CLOSEFILE(%SIMP,DEMO);
```

Chapter 4
SUMMARY AND CONCLUSIONS

4.1 Summary

A MACSYMA based processor for aiding in the systhesis of stiffness matrices for finite element applications has been introduced. It is hoped that the system will significantly reduce the computational effort in the generation of elements, as well as permit investigations which would be intractable by current methods.

It is also hoped that this project has suggested direction for future research and demonstrated the potential of applying MACSYMA to other engineering problems.

4.2 Suggestions for Future work

Some possible areas of future research include the implementation of the extensions listed below. The ramifications of many of these extensions are discussed in section 4.3.

(a) Permit the formulation of consistent mass matrices, load vectors, geometric stiffness terms amd thermal effects (see section 2.9).

(b) Admit second and higher derivatives in the specification of strain components.

(c) Remove the restriction that each degree of freedom must be represented by the same shape function.

(d) Remove the restriction that each node must have the same number of degrees of freedom.

(e) Investigate more efficient methods of performing the symbolic operations.

(f) Investigate criteria and methods to produce better expression optimization in the final matrix template.

(g) Investigate extensions to incorporate material and geometric non-linearities.

(h) Investigate alternative formulations.

(i) Permit the specification and production of additional output formats directly compatible with the calling sequences of a variety of invoking analysis programs.

4.3 Conclusions

It is felt that the current version of ***** has fulfilled the problem statement and design objectives within the limitations and constraints of the operating environment. This is not, however a statement that the problem area has been exhausted. Further research in the areas presented in the section, FUTURE WORK are definitely necessary.

Those areas offering the greatest return in extending the capabilities of the system for the least investment in development costs are to: (a) admit second derivatives in the specification of the stain components; (b) remove the restriction that each degree of freedom must be reperesented by the same shape function and (c) permit the formulation of consistent mass matrices and load vectors. Investigations of the admissibility of material nonlinearities may also prove fruitful.

The implementation of many algorithms merits re-examination. Operations using FOR loops should, whenever feasible, be replaced with vector operations. An examination of the templates generated indicates that attention should be directed towards facilitating the computation through the identification of patterns resulting in the repetition of stiffness coefficients. This technique has been employed in Luft[1], Anderson and Noor[4] and Anderson and Bowen[5].

Futhermore, considerable additional improvements in execution speed may be obtained if futher research is carried out on the optimization of the matrix template. This area represents a significant research effort in itself. Refinement of these output formats could readily permit execution time checks on numerical results. One such check could determine if the determinant of the Jacobian is equal to zero. The symbolic form of the determinant is generated during processor execution and would be imbedded in an appropriate boolean expression.

An important ramification of this research is the demonstration of the potential of MACSYMA in the solution of engineering problems. Though proficiency in its use requires an appreciable overhead to engineers acquainted only with FORTRAN, the rewards are enormous. In addition to the ease and variety of computation possible, the use of MACSYMA permits investigation of general parametric representations heretofore unattainable. Present limitaions in its operation will diminish as additional storage space is made available. There are indications that, through the institution of a virtual memory processor on the computer supporting MACSYMA, a future version will provide a greater alotment of storage space.

SUMMARY AND CONCLUSIONS

In this study, it was found that integration was proportionally the most costly component of the symbolic process. The hybrid symbolic-numerical scheme presented could eliminate this problem, at the expense of a larger amount of numerical computation at execution time. The tradeoffs between the symbolic and hybrid schemes merit further study.

Assessing the present lack of tabulated closed-form formulations of finite element stiffness matrices, it is apparent that ***** can make a significant contribution to the present field of knowledge, permitting display and examination of the closed form expressions.

The impact upon contemporary analysis programs which employ numerical quadrature merits investigation. A sense of the potential is apparent when one considers that these traditional computation techniques require an investment of effort proportional to the number of elements used in each problem to be modeled. Alternatively, ***** need be used only once for each element type. The stiffness matrix so generated is valid for any problem requiring that element type.

FOOTNOTES

[1] Mathlab Group, "MACSYMA Primer" (Cambridge: Mathlab Group, Laboratory for Computer Science, Massachusetts Institute of Technology, 1975), p. 1.

[2] Richard Bogen, "MACSYMA Reference Manual" (Cambridge: Mathlab Group, Laboratory for Computer Science, Massachusetts Institute of Technonogy, 1975), p. 2.

[3] Ibid., p. 2.

[4] Ibid., p. 2.

[5] Ibid., p. 3.

[6] Ibid., p. 125.

[7] Ibid., p. 40.

[8] Ibid., p. 42.

[9] Ibid., p. 81.

[10] Ibid., p. 40.

[11] Ibid., p. 42.

[12] Ibid., p. 81.

REFERENCES

{1} Luft, Rene W., and others. 'Automatic Generation of Finite Element Matrices,' Fifth Conference on Electronic Computation, American Society of Civil Engineers, August 31 to September 2, 1970, pp. 573-94.

{2} Gunderson, Richard H., and Ayham Cetiner. 'A Stiffness Matrix Generator,' Fifth Conference on Electronic Computation, American Society of Civil Engineers, August 31 to September 2, 1970, pp. 595-618.

{3} Taig, I. C.. 'Structural Analysis by the Displacement Method,' English Electric Aviation Ltd., Wharton, England, Rept.SO17, 1961.

{4} Anderson, C. M., and Ahmed K. Noor. 'A Computerized Symbolic Integration Technique for Development of Triangular and Quadrilateral Composite Shallow-Shell Finite Elements,' NASA TN D-8067, December 1975.

{5} Anderson, C. M., and John T. Bowen. 'A Computer Program for Anisotropic Shallow-Shell Finite Elements Using Symbolic Integration,' NASA TM Y-3325, June 1976.

{6} Wong, Andrew Ka-Ching. 'Symbolic Computing,' Numerical and Computer Methods in Structural Mechanics. Ed. Fenves, Perrone, Robinson and Schnobrich. New York: Academic Press, 1973.

{7} Bogen, R., and others. MACSYMA User's Manual. Project MAC, M.I.T, Cambridge, Mass., 1973.

{8} Mathlab Group. MACSYMA Primer. Cambridge: Project MAC, Massachusetts Institute of Technology, 1975.

{9} Desai, Chandrakant S., and John F. Abel. Introduction to the Finite Element Method. New York: Van Nostrand Reinhold Co.,1972.

{10} Zienkiewicz, O. C. . The Finite Element Method in Engineering Science. London: McGraw Hill, 1971.

{11} Lewis, Ellen. 'An Introduction to ITS for the MACSYMA User' Mathlab Memo #3. Cambridge: Mathlab Group, Laboratory for Computer Science, Massachusetts Institute of Technology, 1976.

{12} Dills, Jack, Art Farley and Mary Shaw, eds. CMU PDP-10 Introductory Users Manual. Pittsburgh: Department of Computer Science, Carnegie-Mellon University, 1973.

{13} Jensen, Jarl and Frithiof Niordson. 'Symbolic and Algebraic

REFERENCES

Manipulation Languages and their Applications in Mechanics', Structural Mechanics Software Series, Vol 1. Ed. Perrone and Pilkey. Charlottesville: University Press of Virginia, 1977.

[14] Rubenstein, Moshe F., Structural Systems - Statics, Dynamics and Stability. Englewood Cliffs:Prentice-Hall Inc., 1970.

[15] Cook, Robert D. . 'More About "Artificial" Softening of Finite Elements'. International Journal for Numerical Methods in Engineering. Zienkiewicz and Gallagher, ed., John Wiley & Sons Inc., V11, 8., p. 1334.

Appendix I

OPERATION DETAILS

I.1 INPUT PROTOCOL

Important and unique features of data entry are summarized below:

(a) All input is format free. Input lines are terminated by a semicolon and not by a return. Input may be in upper or lower case.

(b) When typing command lines, depressing the "rubout" or "delete" key deletes the previous character. By typing "ctrl K", the user obtains a copy of the current command line free of any echoed erasures. The two characters ?? delete the whole command line and causes the line number to be redisplayed [4].

(c) In general input is prompted for with a question mark.

I.2 ACCESSING THE SYSTEM

The procedure for gaining access to ***** involves three processes:

(1) Establishing a connection, via the ARPANET, with the MC computer at M.I.T.;

(2) Accessing *****;

(3) Terminating the connection to MC.

I.2.1 Establishing the ARPANET Connection

This procedure is installation dependent and is outlined as it would be enacted on the PDP-10A computer in the Computer Science Department at Carnegie-Mellon University. Specific details of this particular procedure can be found in reference [12].

(1) Login to the computer at CMU, henceforth refered to as the host.

(2) Establish the ARPANET connection with MC I.T. by typing

imp telnet mit-mc

I.2.2 Accessing *****

Details of logging onto MC and of ITS, the operating system on MC, can be found in reference [11]. The user should read sections II.B, and II.G as an absolute minimum. The procedure for initiating execution of ***** is outlined as:

(1) Login to MC

(2) Create a job and load a copy of MACSYMA by typing

:A

(3) MACSYMA will prompt with line C1. Respond with

batch([fstar,cmd,dsk,aklg],on);

This command does not require a carriage return since it is a MACSYMA command. Execution of ***** is initiated.

(4) The logout procedure will depend upon the option chosen, in phase TERMIN, to terminate the system:

If option 3 (TERMINATE THE RUN, THE JOB AND LOGOUT) is chosen, proceed to step (5).

If option 2 (TERMINATE THE RUN AND THE JOB) is selected, control will return to ITS command level (see Lewis[11]). Type

:logout

and proceed to step (5).

If option 1 (TERMINATE THE RUN) is requested, control will return to MACSYMA command level and the user will be prompted for a MACSYMA command with a line of the form " (Ci) ". To exit, type a "ctrl Z" and follow the procedure for option 2.

(5) Return to the host by entering a "ctrl backarrow". (On some terminals this will be a "ctrl underscore" or "ctrl shift O".

I.2.3 Terminating the Connection to MC

This process is also installation dependent. For the host at CMU, the user need only enter

imp close/self

and then logout.

I.3 Additional Notes

(a) Execution of the system may be aborted at any time by entering a "ctrl z". Control will return to ITS command level and the user may restart by typing

:kill

and then follow the procedure starting with step (2) of section I.2.2 or exit by executing the procedure starting with step (5) of that same section.

(b) The Record File (section 2.5.1) may be retrieved for printing at the host by using the file transfer program (FTP) which is documented in reference [12].

(1) While on MC, use the ITS FIND command to determine the file name by typing

:find users;loginname *

where "loginname" is replaced by your MACSYMA login name (see reference [11]). This will display the directory information for all record files which the user has created by executing the system. From the time and date information displayed, the user can determine the desired file. The file name will be your login name followed by an integer. The file with the greatest integer is the most recently created.

(2) On the host enter

```
r ftp
host mit-mc
user users
retr users;loginname integer
quit
quit
```

where "integer" is the value displayed by the FIND command. This will create a file on the user's directory on the host, called "USERS" which will contain the contents of the record file. The file may be dumped to the line printer or examined using one of the editors. The file on MC is left untouched and may be erased using the ITS DELETE command.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER R-78-103	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Symbolic Generation of Finite Element Stiffness Matrices		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER R-78-103
7. AUTHOR(s) Alan R. Korncoff, Steven J. Fenves		8. CONTRACT OR GRANT NUMBER(s) N0014-76-C-0354
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-064-536
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Structural Mechanics Program, Arlington, VA 22217		12. REPORT DATE January, 1978
		13. NUMBER OF PAGES 109
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research, CMU Campus Representative, 5000 Forbes Avenue, Pittsburgh, PA 15217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution unlimited for any purpose of the United State Government

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Finite Element Analysis

Symbolic Computing

use five asterisks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A symbolic processor, ***** (pronounced "five star"), to assist in the generation of stiffness matrices for finite elements, based on a recently developed symbolic processor, is presented. Operations are performed upon element characteristics and material properties in symbolic form to produce a "matrix template," consisting of the algebraic expressions generated for the stiffness coefficients as functions of the problem parameters

Insert page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) in literal form. The template may be evaluated for a given element by binding these symbolic forms to the numerical values associated with a specific element. The evaluation process is further facilitated by permitting specification of a variety of output formats for the resulting matrix template. Required input is minimized by automatically synthesizing the constituent matrices of the formulation from user-supplied specifications of shape functions, material properties and stress-strain relationships, all in symbolic notation.

The processor, written in MACSYMA, is highly interactive providing prompts for user input, enumeration of available program options, and extensive on-line assistance. The user may input a "?" in place of a prompted input to request instructional text. The file handling capabilities of MACSYMA are utilized to retain a complete record of each program run. These records facilitate the handling of diagnostics, assist in further processing and permit the generation of statistics valuable for system development. Error checking is accomplished through semantic checks built into the program functions and syntactic checks performed within the MACSYMA operating environment.

A partial list of user input includes:

1) Method Selection - Isoparametric or generalized coordinate formulations. 2) Element Parameters - Number of nodes, number of degrees of freedom per node and related terms. 3) Material Properties - This matrix may be selected from a library of standard forms (e.g. plane stress, plane strain) or supplied by the user. 4) Strain Specification - Components are entered in a user-oriented calculus notation (e.g. $\partial u/\partial x$ is input as D(u,x)). 5) Shape Functions - Shape functions may contain trigonometric functions and a large class of intrinsic functions as well as polynomial terms. 6) Output Control Specification - A description of the output format of the generated matrix template.

Possible output forms include a tabular display of the matrix coefficients in symbolic form and the coefficients in FORTRAN card image format.

Background material includes: The objective of this study; The derivation of the stiffness matrices; A summary of previous research; A brief description of MACSYMA.

Details of the implementation of ***** cover: The design objectives; Details of the algorithms used and how they were implemented; A description of ***** and its limitations.

Sample runs include the formulation, using both the isoparametric and generalized coordinate methods, of the stiffness matrices for a: Bar element with constant cross-sectional area; Bar element with linearly varying cross-sectional area; Constant Strain Triangle with uniform thickness; Four Node Quadrilateral.

Conclusions are drawn and recommendations for future work are made. Appendix I contains notes on operating and accessing the processor.

IED

-78